Honeywell

UTILITY

SERIES 60 (LEVEL 66)/6000

GCOS

SOFTWARE

Honeywell

UTILITY

SERIES 60 (LEVEL 66)/6000

GCOS

SUBJECT:

General Description, Usage, Functions, and OWN CODE Capability of Utility Software System

SPECIAL INSTRUCTIONS:

This manual replaces Utility, Order No. BQ66 for Series 6000 System users. Order No. BQ66 must be used by Series 600 System users and by Series 6000 System users who are on prior software releases.

SOFTWARE SUPPORTED:

Series 60 Level 66 Software Release 2
Series 6000 Software Release H

INCLUDES UPDATE PAGES ISSUED AS ADDENDUM A, DATED JANUARY 1975.

DATE:

April 1974

ORDER NUMBER:

DD12, Rev. 0

PREFACE

Utility is a generalized software system that provides a wide variety of peripheral storage device processing capabilities. This manual describes the use of Utility with the Series 60 Level 66 and Series 6000. The Series 60 Level 66 is hereafter referred to as the Series 60. The information in this manual refers to both the Series 6000 and Series 60 unless otherwise specifically stated.

The manuals referenced in this manual are:

Series 60 (Level 66)/6000 Control Cards, Order No. DD31
Series 60 (Level 66)/6000 File and Record Control, Order No. DD07

© 1974,1975, Honeywell Information Systems Inc. File No.: 1753, 1P53

FUNCTIONAL LISTING OF PUBLICATIONS
for
SERIES 60 (LEVEL 66) and SERIES 6000 SYSTEMS

FUNCTION	APPLICABLE REFERENCE MANUAL TITLE Series 60 (Level 66)/Series 6000:	ORDER NO.
Hardware reference:		
Series 60 Level 66 System	Series 60 Level 66 Summary Description	DC64
Series 6000 System	Series 6000 Summary Description	DA48
DATANET 355 Processor	DATANET 355 Systems Manual	BS03
DATANET 6600 Processor	DATANET 6600 Systems Manual	DC88
Operating system:		
Basic Operating System	General Comprehensive Operating Supervisor (GCOS)	DD19
Job Control Language	Control Cards Reference Manual	DD31
Table Definitions	System Tables	DD14
I/O Via MME GEINOS	I/O Programming	DB82
System initialization:		
System Startup	System Startup	DD33
System Operation	System Operating Techniques	DD50
Communications System	GRTS/355 and GRTS/6600 Startup Procedures	DD05
Communications System	NPS Startup	DD51
DSS180 Subsystem Startup	DSS180 Startup	DD34
Data management:		
File System	File Management Supervisor	DD45
Integrated Data Store (I-D-S)	I-D-S/I Programmer's Guide	DC52
Integrated Data Store (I-D-S)	I-D-S/I User's Guide	DC53
File Processing	Indexed Sequential Processor	DD38
File Input/Output	File and Record Control	DD07
File Input/Output	Unified File Access System (UFAS) (Series 60 only)	DC89
I-D-S Data Query System	I-D-S Data Query System Installation	DD47
I-D-S Data Query System	I-D-S Data Query System User's Guide	DD46
Program maintenance:		
Object Program	Source and Object Library Editor	DD06
System Editing	System Library Editor	DD30
Test system:		
Online Test Program	Total Online Test System (TOLTS)	DD39
Test Descriptions	Total Online Test System (TOLTS) Test Pages	DD49
Error Analysis and Logging	Honeywell Error Analysis and Logging System (HEALS)	DD44
Language processors:		
Macro Assembly Language	Macro Assembler Program	DD08
COBOL-68 Language	COBOL	DD25
COBOL-68 Usage	COBOL User's Guide	DD26
JOVIAL Language	JOVIAL	DD23
FORTRAN Language	FORTRAN	DD02
Generators:		
Sorting	Sort/Merge Program	DD09
Merging	Sort/Merge Program	DD09

FUNCTION	APPLICABLE REFERENCE MANUAL	
	TITLE Series 60 (Level 66)/Series 6000:	ORDER NO.
Simulators:		
DATANET 355/6600 Simulation	DATANET 355/6600 Simulator	DD32
Service and utility routines:		
Loader	General Loader	DD10
Utility Programs	Utility	DD12
Utility Programs	UTL2 Utility Routine (Series 60 only)	DC91
Media Conversion	Bulk Media Conversion	DD11
System Accounting	Summary Edit Program	DD24
FORTRAN	FORTRAN Subroutine Libraries	DD20
FNP Loader	DATANET 355/6600 Relocatable Loader	DD35
Service Routines	Service Routines	DD42
Software Debugging	Debug and Trace Routines	DD43
Time Sharing systems:		
Operating System	TSS General Information	DD22
System Programming	TSS Terminal/Batch Interface	DD21
System Programming	TSS System Programmer's Reference Manual	DD17
BASIC Language	Time Sharing BASIC	DD16
FORTRAN Language	FORTRAN	DD02
Text Editing	Time Sharing Text Editor	DD18
Remote communications:		
DATANET 30/305/355/6600 FNP	Remote Terminal Supervisor (GRTS)	DD40
DATANET 355/6600 FNP	Network Processing Supervisor (NPS)	DD48
DATANET 700 RNP	RNP/FNP Interface	DB92
Transaction processing:		
User's Procedures	Transaction Processing System User's Guide	DD41
Handbooks:		
System-operator communication	System Console Messages	DD13
Pocket guides:		
Control Card Formats	Control Cards and Abort Codes	DD04
FORTRAN	FORTRAN Pocket Guide	DD82

CONTENTS

CONTENTS (cont.)

ILLUSTRATIONS

SECTION I

INTRODUCTION

PURPOSE

Utility is a system providing peripheral storage device processing capabilities. It permits copying, copying with merge capability, copying with resequencing capability, comparing, positioning, and printing.

Utility is used both for operational and debug purposes. It resides in system storage and is called by the user through the General Comprehensive Operating Supervisor (GCOS) (unconditionally by the $ UTILITY card or conditionally by the $ ABORT card).

CAPABILITY

Utility processes magnetic tapes (logically or physically), linked disk files (logically only), or random disk files as specified by user-supplied control cards. Utility system capabilities are:

- Copy (COPY) a specified number of files and/or records from first file code to second.
- Copy (MCOPY) the information from a specified number of files and/or records from first file code to second without any separating or terminating labels or file marks (17 octal).
- Print (DUMP, ADUMP, AADUMP, DDUMP) a specified number of files and/or records.
- Compare (COMP) a specified number of files and/or records from first file code to second.
- Position a specified storage device:
    1. Rewind (REW or RWD).
    2. Skip (SKIP) a specified number of files and/or records on tape or disk.
    3. Backspace (BKSP) a specified number of files and/or records on tape or disk.
- Provide user with capability (OWN CODE) for adding his own instructions for a Utility run.
- Repeat (RPT) the next m parameters n times.
- Copy (RSAVE) a random file to magnetic tape.
- Copy (RREST) a file from magnetic tape to a random file.
- Copy (RCOPY) a random file to another random file.

SECTION II

GENERAL USAGE

UTILITY CONTROL CARDS

The following are control cards associated with Utility:

1	8	16	
$	UTILITY	(DUMP/NDUMP)	Utility call card(s)
$	ABORT	(None)	
$	(File)	(Variables)	Standard GCOS file control card(s) such as $ TAPE, $ TAPE9, and $ FILE.
$	QUTIL	(Variables)	Describes Utility processing options.
$	FFILE	(Variables)	Describes nonstandard file control blocks.
$	FUTIL	(Variables)	Describes functions to be performed by Utility.
$	ETC	(Variables)	Continuation card which can be used if parameters do not all fit on a $ FFILE, $ FUTIL, or $ QUTIL card.

STANDARD OPTIONS (Assumes System Standard Format)

The following standard options are assumed for a particular file code.

Block Size:

Logical record processing: 320 words.

Physical record processing: length of available storage for all but the Compare function. Because the Compare function must have two records in memory at the same time, it uses half of the available storage for each record.

Number of Buffers:

Logical record processing: Two buffers per file code.

Physical record processing: Two buffers for Compare; all other functions, only one common buffer.

Record Form:

Variable-length records

Labels:

Standard File and Record Control labels

Block Serial Numbers:

Assumed

Density:

High[1]

Mode:

Binary

Retention Period:

0

Mode of Processing:

Logical record processing

NONSTANDARD OPTIONS

Nonstandard options are specified to Utility by means of a $ FFILE card. This card is in the following format:

1	8	16	72
$	FFILE	file code, parameters	

[1] A density setting parameter on a $ TAPE control card controls the density setting for the MTU0400, MTU0500 (Series 60 only) or MTS500 magnetic tape subsystem. If this optional field is not utilized, the system default high density setting is selected.

A file code is a two-character alphanumeric code assigned to the file. This is the same file code assigned by the user on the GCOS file card to uniquely identify a particular file. Parameters must be separated by commas.

The parameters for nonstandard options processed by Utility are:

NSTDLB or NLABEL	Do not create or verify standard labels	If the PHYREC option is used: for an input file--these are unnecessary. For an output file--MODBCD and LODENS are ignored if ASIS is specified on a $ QUTIL card preceding the $ FUTIL card.[3]
MODBCD or MBCD	Set recording mode to BCD	
LODENS[1] or LDENS	Set device to low density	
ASA9[2]	Set recording mode to ASA 9-track tape format. The RT9 command is used for reading and WT9 is used for writing by File and Record Control. Applicable to 9-track magnetic tape only.	
BUFSIZ/n	Set buffer size to n. Limit is 4095 words or less, depending upon storage available.	Ignored for an input file if PHYREC option is being used for that file. Also ignored for an output file if it is a tape file being made by a physical (PHYREC) copy function.
FIXLNG/n or FXXX	Records are fixed length, n words long.	
NOSRLS or NSER	Do not use block serial numbers	
RETPER/n or RXXX	Place the retention period n into the output label. The upper limit for n is 999. If a retention period greater than 999 is specified, 999 is used. Applicable only to labeled output magnetic tape files.	

[1] For ASA magnetic subsystems, the LODENS parameter should be used for low density tapes. For the MTU0400, MTU0500 (Series 60 only) or MTS500 magnetic tape subsystems, the density should be made known to Utility through the Density parameter on the $ TAPE control card.

[2] The ASA9 parameter should not be used for normal 9-track tape handling. It is used only when the special ASA9 recording format is required.

[3] For the MTU0400, MTU0500 (Series 60 only) or MTS500 magnetic tape subsystem, a density setting field on the $ TAPE control card for the output file is ignored if the ASIS option is specified.

NDATE For a COPY function of labeled magnetic tape to labeled magnetic tape, use the current (new) date as the creation date for the output file rather than copying the creation date from the input file. This option can be used only on an output $ FFILE card.

MLTRL Specifies that input is an unlabeled file on multiple reels. A CE message is given to the operator at a standard EOF. If the MLTRL option is not used, Utility does not read more than one reel of an unlabeled file.

MLTFIL Specifies multifiles. It should be used if there is a possibility that an output magnetic tape file may exceed one reel. This prevents the creation of an illegal file (being both multifile and multireel).

PHYREC Utility determines the mode (except ASA9, which must be specified), density, and labeling conditions of the input magnetic tape file. Physical tape records, limited in size only by available storage, are processed (see PHYREC, below) by means of DCWs generated within Utility. PHYREC is applicable only to magnetic tape files.

RANDOM/n This input file is a random disk file that can be DUMPed, DDUMPed or SKIPed. The dump or skip is on the basis of fixed-length records only, n words long (n cannot be larger than 1280 words). If n=0, record size is assumed to be 64 for disk storage files. Processing of a given RANDOM file must be completed on one $ FUTIL card because all open data files are closed at the completion of a $ FUTIL card. Processing of a RANDOM file must also be completed before processing of another file is started. (See Section III for an example of the use of the random disk file dump/skip capability.) The random parameter can also be used for dumping a linked file. This can be done by declaring the linked file random on its GCOS control card and on Utility's FFILE card when dumping the file.

Note that n is written as an unsigned decimal integer. Parameters not mentioned above are listed as illegal parameters and then ignored.

The nonstandard options specified on a $ FFILE card remain in effect for that file code for the rest of the activity unless changed by another $ FFILE card, which always (even if it contains nothing but an illegal parameter) completely resets Utility $ FFILE tables for that file code. A $ FFILE card must precede a $ FUTIL card if the nonstandard options it specifies for a file are to be in effect for the functions specified on that $ FUTIL card. This is because Utility processes its control cards one at a time. The Utility routine permits a maximum of 24 $ FFILE control cards per activity.

PHYSICAL RECORD PROCESSING

Physical record processing (PHYREC) provides the following capabilities not available with logical record processing.

1. It allows processing of magnetic tape input about which nothing is known (for example: mode and record form).

2. It allows processing of magnetic tape records larger in size than 4095 words, the maximum size for logical record processing.

3. It allows processing of mixed mode, mixed density magnetic tape files. Mixed mode is handled within files; density (and labeling conditions) may change only from file to file.

PHYREC need not be specified for an output file for a physical copy. The COPY function, when the PHYREC option has been specified for the input file, causes physical copying to the output file. The output file is treated in a physical mode for all subsequent FUTIL functions on the same $ FUTIL card. It is, however, treated logically for any subsequent $ FUTIL cards unless it is explicitly set for physical record processing by a ($ FFILE) PHYREC option before the subsequent $ FUTIL cards.

The user should be aware of the fact that during physical record processing, block serial numbers are considered only as data words if they appear on the input file. If a file with block serial numbers is being copied physically and any input records are skipped either by the SKIP function or the IGNORE option, the corresponding block serial numbers will also be missing in the output. (Use of the RESEQ option on a $ QUTIL card avoids these problems.) Also, if NSTDLB option is used with physical record processing, a standard label, followed by an EOF mark (octal 17), is treated as one file.

UTILITY FUNCTIONS

Utility functions are defined by the $ FUTIL card as follows:

1	8	16	72
$	FUTIL	filecode 1,filecode 2,options	

The operand field contains the file code of each of the two files (maximum per $ FUTIL card) being serviced. Utility creates a file control block and, for logical record processing, reserves two buffers for each file code. These file control blocks are set to standard options unless previously modified by $ FFILE cards. If two file codes are not needed, a comma must be used in place of the missing file code. Multiple $ FUTIL cards are permitted within an activity. The RPT function and all functions to be repeated must be on a single $ ETC card that follows and continues a $ FUTIL card.

The functions permitted by Utility are as follows:

Option	Meaning
COPY/M/	Copy M from first file (filecode 1) to second file (filecode 2).
MCOPY/M/	Copy M from first file (filecode 1) to second file (filecode 2) without separating or terminating labels or file marks (17 octal).
COMP/M/	Compare M from first file (filecode 1) to second file (filecode 2).
SKIP/M,N/	Skip M on first file (filecode 1) and N of second file (filecode 2).
BKSP/M,N/	Backspace M on first file (filecode 1) and N of second file (filecode 2).
DUMP/M,N/	Dump M from first file (filecode 1) and N from second file (filecode 2) in octal and BCD equivalent.

ADUMP/M,N/
Dump M from first file (filecode 1) and N from second file (filecode2) in octal and it's ASCII equivalent (applicable only to time sharing created files, type 5 or 6, and to ASCII files under physical record processing).

RPT/mP,nT/
Repeat the next m functions n times.

AADUMP/M,N/
Dump M from first file (filecode 1) and N from second file (filecode 2) in ASCII representation only (applicable only to time sharing created files, type 5 or 6, and to ASCII files under physical record processing).

DDUMP/M,N/
Dump M from the first file (filecode 1) and N from the second file (filecode 2) in BCD representation only.

RWD (or REW)/filecode 1,filecode 2/
Rewind file filecode 1 and file filecode 2.

HOLD/filecode 1,filecode2/
Do not close files filecode 1 and filecode 2 at the completion of the functions specified on this $ FUTIL card (see HOLD definition, Section IV).

RSAVE/1F/
Copy one random file (filecode 1) to a magnetic tape (filecode 2).

RREST/1F/
Copy one file from magnetic tape (filecode 1) to a random file (filecode 2).

RCOPY/1F/
Copy one random file (filecode 1) to random file (filecode 2).

The M represents the number of files (#F) and/or number of records (#R) from filecode 1, and the N represents the number of files and/or records from filecode 2. (Neither M nor N can be zero.) These parameters can be specified, for example, as 3F4R, 4F, or 5R. (If an end-of-file is encountered during a record countdown, the record count is considered exhausted.) File codes referred to here as filecode 1 and filecode 2 can be designated by any two alphanumeric characters.

UTILITY PROCESSING OPTIONS

Optional courses of action that can be taken by Utility during its processing can be specified on the $ QUTIL card.

1	8	16	72
$	QUTIL	Options	

The options that can be specified on the $ QUTIL card are as shown below. (If an option is not specified, the underlined option is assumed.)

Option — Meaning

EOF/ALL

Count all file marks (except octal 75, 76) as file delimiters. (Nonstandard file marks are still copied to output during a copy function even though they are now treated as file delimiters.)

Count only octal 17 file marks as file delimiters.

ASIS

Copy to output tape in same mode, density, and labeling condition as input tape. Applicable only to PHYREC processing.

Copy according to nonstandard options indicated on output $ FFILE card; where none, copy according to standards.

NOTE: For the MTU0400, MTU0500 (Series 60 only) or MTS500 Tape Subsystems, DEN5 must be specified on the input $ TAPE card when doing an ASIS copy of a 556 bpi input tape file. Otherwise, an 800 bpi output tape may result.

CMPERR/n

n is the number of compare errors which will be accepted. Utility terminates with a fatal error on the n + 1 compare error. n can range from 0 to 262,143. If the end of the Utility activity is reached before n + 1 compare errors are encountered, the presence of one or more compare errors still causes a fatal error.

Terminate activity with fatal error when 50 compare errors have been encountered.

NBYPSS

Do not bypass noise records. The Exception Processing override option, which provides no noise record test, is used. Applicable to input magnetic tape file only.

The NBYPSS option should be used if there is a possibility that the tape to be read contains any 1- or 2-word physical records. If the option is not used, 1- or 2-word physical records having bad parity are treated as noise records and so bypassed.

Whenever the NBYPSS option is used, the density and mode of the input tape must be made known to the Utility program in advance. The Utility program cannot determine the density and mode while the program is in the physical record processing mode, as is usually the case.

Bypass noise records.

USE/n or IGNORE/n n is the number of input nonrecoverable parity errors accepted (with the appropriate USE or IGNORE action taken) before the activity is terminated with a fatal error.

Terminate activity with fatal error when an input nonrecoverable parity error is encountered.

The following table summarizes action taken for the USE and IGNORE options on nonrecoverable input parity errors until count n is exceeded:

	Directives		
Option	Dump	Copy	Compare
Use	Dump complete physical record with error message. Continue.	Dump physical record on P* with error message. Copy and continue.	Dump physical record on P* with error message; compare and continue.
Ignore	Print error message only. Continue.	Dump physical record on P* with error message; do not copy; continue.	Dump physical record on P* with error message; terminate activity immediately with a fatal error.

The error message on P* is as follows:

```
FILE-XXXX BLOCK-XXXX FILE CODE XX NONRECOV. PARITY ERROR
```

For input nonrecoverable parity errors: the operator is always bypassed during physical record processing; the operator is bypassed during logical record processing if a USE or IGNORE option is used.

It is not recommended that the USE options be used for logical record processing of variable-length records. If the record-size field of one of the record-size control words is in error, the input record cannot be unblocked correctly; and the run is aborted.

Option Meaning

RESEQ During a physical record COPY or MCOPY, reserialize the output file (build new block serial numbers).

Do not reserialize but copy the block serial numbers from the input to the output.

If during the course of resequencing COPY or MCOPY, an input block number is out of sequence, the following message is put to P*:

```
*BSN ERR*. EXPECTED XXXXXX, FOUND XXXXXX, RESEQ'D TO XXXXXX
```

In addition, the block is dumped to P*, the expected block number counter is reset to the block number of this block, and the block itself is given the next sequential output block sequence number before it is copied to the output file.

This option is useful not only for building a reserialized copy of an existing tape file with block serial numbers out of sequence, but also for building a serialized output file in the physical record processing mode from pieces of other files. For example, an MCOPY of three files on one tape in the physical record processing mode would result in one file on the output, the one file having block serial numbers out of sequence (input block serial numbers copied to output) unless the RESEQ option were used.

If Block Serial Numbers are used, the RESEQ option must be selected for a physical copy of multi-reel files to avoid the BLK. SERIAL SEQUENCE IN ERROR message.

Option	Meaning
USER	The user has supplied subroutine(s) to modify files during a UTILITY run.

The options specified on a $ QUTIL card remain in effect for the rest of the activity, with the following exception: three $ QUTIL options can be reset in the same activity by use of another $ QUTIL card. These are:

USE, IGNORE, and CMPERR

A $ QUTIL card must be placed before a $ FUTIL card if the processing options that it defines are to be in effect for the functions specified on the $ FUTIL card.

SECTION III

SPECIFIC USAGE

CALLING IN UTILITY

Utility can be referred to as a separate activity or as a subactivity of a particular job. The $ UTILITY control card is needed to call Utility as a separate activity.

1	8	16
$	UTILITY	Options

The following options can be specified in the operand field. (If an option is not specified, the underlined option is assumed.):

DUMP: Dump all of slave core if the Utility activity is aborted.

NDUMP: Program registers, upper SSA, Utility's File Control Block, and slave program prefix are dumped if the Utility activity is aborted.

When the System Input Collector encounters this card, it opens a U* file. All subsequent Utility control cards, up to the next $ control card indicating a new activity, are placed on that file. The U* file then becomes the input file for Utility.

The $ ABORT card (used instead of the $ UTILITY card) calls Utility as an abort subactivity.

1	8	16
$	ABORT	(Not used)

The $ ABORT control card begins definition of a subactivity to be initiated only if the activity immediately preceding it is aborted. If this occurs, the $ FFILE, $ FUTIL, $ QUTIL, and $ ETC cards (the only cards following the $ ABORT card that are written by the System Input Collector on the U* file) are read in and processed by Utility.

OTHER CONTROL CARDS

A particular Utility activity may require a request for extra storage. (See Utility Storage Requirements in this section.) This can be done by placing a $ LIMITS card immediately behind the $ UTILITY control card. The $ LIMITS card can also be used to change the Utility SYSOUT line limit, normally 10,000.

For further descriptions of the control cards, see the Control Cards Reference Manual.

UTILITY STRUCTURE

Utility is a slave program that provides two modes for processing tapes. The processing can be either by logical records (normally) or by physical records (by designating PHYREC on the input $ FFILE card). Only magnetic tape files can be handled by PHYREC processing. Logical records are read by means of the GET routine and written by the COPY routine. Physical records are read by the READ routine and written by the WRITE routine. These routines are explained in the File and Record Control reference manual.

During initialization, storage availability is checked and buffers are set up. The U* file, previously written by System Input with Utility control cards, is opened.

Only $ FFILE, $ FUTIL, $ QUTIL, and $ ETC cards are processed. In the case of a $ FFILE card, only options listed in the "Nonstandard Options" section are processed; all others are ignored. In the case of a $ FUTIL card, file code parameters are checked. If a $ FFILE card was present for the particular file code, the file control block is set to the desired options. If no $ FFILE card was present, standard options are placed in the file control block. If Utility generates two file control blocks, FCB1 refers to the first file code and FCB2 to the second. Initialization for each $ FUTIL card also includes reinitialization of "running" file and record counts. The $ FUTIL card is then scanned for Utility functions, and as each function is recognized (a second slant encountered), this option is processed. All parameters in the $ FUTIL card must conform to those specified in "Utility Functions", Section II.

I/O activities within Utility are performed with standard File and Record Control routines. When all functions specified on a $ FUTIL card are completed, files are closed (without rewind and without lock) unless the Hold option was specified. If Hold was used, the file control block for that particular file code is not closed. The Utility activity terminates when an end-of-file is encountered on the U* file.

DECK SETUPS

Particular attention should be paid to the description of nonstandard files as described under "Nonstandard Options" in Section II. Note that Utility recognizes only nonstandard options; entries such as STDLBL are ignored.

An example of a Utility deck setup is as follows:

```
1      8         16
$      SNUMB
$      IDENT
$      UTILITY
$      FUTIL     BD,,RWD/BD/,DUMP/1F/
$      TAPE      BD,X1R,,1234
$      ENDJOB
```

In this example, file BD is rewound and one file is dumped from it. Note that two file code parameters must be specified on $ FUTIL cards, although one can be a null field. In subsequent examples, $ SNUMB, $ IDENT, and $ ENDJOB cards are not listed but are assumed present.

```
1      8         16
$      UTILITY
$      LIMITS    ,16384
$      FFILE     F1,PHYREC,LODENS
$      FUTIL     F1,F2,RWD/F1,F2/,COPY/10F/,RWD/F1,F2/,
$      ETC       COMP/10F/,RWD/F1,F2/,DUMP/,10R/,RWD/F2/
$      TAPE      F1,X1D,,1234
$      TAPE      F2,X2D,,,,COPYFILE
```

The above deck setup requests Utility to copy physically (rather than logically) 10 files from a tape of possibly unknown composition to a labeled, high density, binary tape. Both tapes are then to be verified by comparison, and 10 records are to be dumped from the second tape (F2). The 16,384 words of storage requested permit comparison of physical (tape) records of up to 4224 words in size or copying and dumping physical tape records of up to 8448 words. See "Utility Storage Requirements", in this section.

```
1      8         16
$      UTILITY
$      QUTIL     EOF/ALL
$      FUTIL     F1,,RWD/F1/,DUMP/99F/,RWD/F1/
$      TAPE      F1,X1R
```

This deck setup asks for a dump of 99 files, all file marks (except 75 and 76) to be considered as file delimiters. If a partial label[1] is encountered, the dump is terminated. By specifying an unusually large number of files in this manner, it is possible to process any produced labeled tape (logical and physical record processing) or unlabeled tape (physical record processing only) without knowing its precise number of files. Utility stops any processing function and proceeds to the next one upon encountering a partial label.

When a partial label is encountered, the tape is backspaced over the partial label and the following message is sent to P*:

```
FILE CODE xx FILE # nn  RECORD #1
IS END OF DATA LABEL. FUNCTION ENDED.
```

The partial label itself is counted as a new file; therefore, the number of useful data files on the device is nn - 1.

```
1      8         16
$      UTILITY
$      FUTIL     F1,F3,RWD/F1,F3/,COPY/3R/,HOLD/F3/
$      FUTIL     F2,F3,RWD/F2/,COPY/1R/,HOLD/F3/
$      FUTIL     F1,F3,RWD/F1/,SKIP/4R/,COPY/1F/
$      TAPE      F1,X1D,,1234
$      TAPE      F2,X2D,,4321
$      TAPE      F3,X3D,,,,OUTFILE
```

The preceding deck setup illustrates the HOLD function. In this example, it is assumed that the fourth logical record of a standard magnetic tape file, F1, is to be changed and that a new copied file, F3, will be created. The corrected record to be copied is on F2. Initially, three records are copied from file F1 to file F3. F3 is held open with the HOLD function, and the new fourth record is copied from file F2. File F3 is again held open with the HOLD function and remainder of file F1 is copied to F3. The HOLD must be used to prevent output file, F3, from being closed before completion of the copying required to make up the logical file (see HOLD definition, Section IV).

[1] A partial label is written by File and Record Control on magnetic tapes as an end-of-information banner. It is distinguished from a standard label by the fact that it has zeros in words 5 through 10.

HOLD can also be used for holding within an input logical file. It could not, however, be used in the above example for F1 at the end of the first $ FUTIL card. This is because two FCBs would have been kept open, tying up both of Utility's FCB's. No new file (F2) could then be introduced.

The following control card setup illustrates the use of the MCOPY function to merge three files from multiple inputs into a single file without intervening labels or file marks (octal 17).

```
1      8       16
_____________________________
$      UTILITY DUMP

$      TAPE    F2,X2D,,1234

$      TAPE    F3,X3D,,2345

$      TAPE    F4,X4D,,3456

$      TAPE    F5,X5D,,,,ONEFIL

$      FUTIL   F2,F5,REW/F2,F5/,MCOPY/1F/,HOLD/F5/

$      FUTIL   F3,F5,REW/F3/,MCOPY/1F/,HOLD/F5/

$      FUTIL   F4,F5,REW/F4/,MCOPY/1F/,REW/F5/,DUMP/,1F/
```

In this setup one file (F5) will have been created from the first file of F2,F3 and F4, without intervening labels or file marks. The newly created file F5, is then dumped using the DUMP option.

The following deck setup illustrates the use of Utility's random file SKIP/DUMP capability.

```
1      8       16
_____________________________
$      UTILITY

$      FFILE   AB,RANDOM/10

$      FFILE   BC,RANDOM/0

$      FUTIL   AB,BC,SKIP/5R/,DUMP/5R,1F/

$      FILE    AB,A1R,3R

$      FILE    BC,B1R,2R
```

In this deck setup, a 3-link random access file, AB, is to be skipped down by five 10-word records. Then, its next five 10-word records are to be dumped to P*. Finally, a 2-link random access file, BC, is to be dumped to P* by 64-word records. (See the $ FFILE parameter, RANDOM/n, in Section II.)

The following deck setup illustrates the use of the Utility RPT capability.

```
1      8       16
______________________________

$      UTILITY

$      FUTIL   FC,FD,RWD/FC,FD/,

$      ETC     RPT/3P,2T/,COPY/10R/,SKIP/10R/,COPY/10R/,

$      ETC     REW/FC,FD/,DUMP/1F,1F/
```

In this example, records 1 through 10, 21 through 40, and 51 through 60 are copied from file FC to file FD. Both files are then rewound. The RPT and all repeat functions must be contained on one $ ETC control card that is a continuation of a $ FUTIL control card. Additional functions (not repeated) can be continued (as indicated) on $ ETC control cards. See Section IV for a definition of RPT.

The following deck setup illustrates the use of the Utility ADUMP (or AADUMP) capability.

```
1      8       16
______________________________

$      UTILITY

$      FUTIL   FC,,ADUMP/1F/

$      PRMFL   FC,R,S,File String
```

The time sharing created linked file FC is dumped in both octal and its ASCII equivalent.

The following deck setup illustrates the use of Utility's AADUMP physical record processing capability.

```
1      8       16
______________________________

$      UTILITY

$      TAPE    IN,X1D,,OLDTAPE

$      FFILE   IN,PHYREC

$      FUTIL   IN,,AADUMP/1F/
```

OLDTAPE is in ASCII format. The tape is dumped in its ASCII representation, using physical record processing.

Upon termination of an MCOPY function, the following message is also placed on P*.

TOTAL RECORDS MCOPIED xxxxxx

The record count xxxxxx shows the total number of records MCOPIED to the output.

Example: If the function was MCOPY/3F/ and the three files contained 300, 300, and 302 records respectively, the record count is 902 records.

COMP Function--No Errors. If there were no compare errors and the COMP function was terminated by record countdown, the following message is placed on P*:

COMPARED xxxxxx FILES xxxxxx RECORDS

If there were no compare errors and the COMP function was terminated by an end-of-file having been reached, the following message is placed on P*:

COMPARED xxxxxx FILES. xxxxxx RECORDS IN LAST FILE

COMP Function--With Errors. If there were one or more compare errors when the compare function terminated, one of the following messages is placed on P* and the activity is terminated with a fatal error.

COMP ERROR COUNT xxxxxx

COMPARE COMPLETED. ERROR COUNT xxxxxx

or if the maximum number of compare errors allowed is exceeded, the compare function is aborted and the following message is used:

COMPARE ABORTED. ERROR COUNT xxxxxx

RSAVE, RREST, or RCOPY More or Less Than One File

An attempt to use the RESAVE, RREST, or RCOPY functions for either more or less than one file at one time, results in the following message:

$$\text{UTILITY} \left\{ \begin{matrix} \text{SAVES} \\ \text{RESTORES} \\ \text{RCOPIES} \end{matrix} \right\} \text{ONLY A FILE}$$

Printout in Compare Function

Every specification of a COMP function results in the following printout that indicates that data on FILE (code) xx is to be compared with data on FILE (code) yy.

COMPARE - xx VS yy

Printout in Backspace Function

Every specification of a BKSP function results in the following printout if the loadpoint is encountered.

```
FILE CODE xx LOADPOINT ON BKSP
```

The following printouts indicate that there was a noncomparison and they identify by number the two files (mm) and two records (nn) which do not compare.

```
FILE # mm -- RECORD # nn, AND FILE # mm -- RECORD # nn

RECORDS DO NOT COMPARE
```

Noncomparison of Data Records. The reason for noncomparison is shown by either of two printouts. The following printout indicates that the two records just compared were not the same size. When sizes do not compare, the COMP ERROR COUNT is incremented by one and no further check is made on the two records. The record sizes are printed, followed by a dump of the two unequal records. The printout is as follows:

```
SIZES DIFFERENT xxxxxx VS yyyyyy
```

where:

xxxxxx and yyyyyy are the sizes of the two records

```
FILE CODE (#1) (DUMP OF FIRST BLOCK)

FILE CODE (#2) (DUMP OF SECOND BLOCK)
```

The following printout indicates that the two compared records are the same size but the data does not compare. When the data does not compare, the following line is printed and the COMP ERROR COUNT is incremented by one for each noncomparison within the two records. The printout is:

```
#### aaaaaaaaaaaa bbbbbbbbbbbb ccccccdddddd
```

where:

```
#### = the word number within the record where the
       noncompare occurred
a--a = the octal representation of the data word
       from file code xx
b--b = the octal representation of the data word
       from file code yy
c--c = the BCD equivalent of the data word
       from file code xx
d--d = the BCD equivalent of the data word
       from file code yy
```

Noncomparison of End-of-File Marks. The following printouts indicate that the EOF marks are not aligned. They specify which unit first encountered the EOF. The other unit is then aligned from a matching EOF before comparisons are resumed. If the number of files to be compared is exhausted during this positioning, comparison is terminated. The printout is either

```
EOF 1ST UNIT ONLY
```

or

```
EOF 2ND UNIT ONLY
```

When the EOF marks are aligned but do not compare, the second device is positioned until either the file marks agree or the number of files or records to be processed is exhausted. The printout for noncomparing file marks is:

```
END FILE MARK - xx VS yy
```

where:

xx and yy are the disagreeing file marks

Printout for Physical End-of-File

While processing files on a random access device, a physical end-of-file condition may be encountered. Utility treats this as though a standard EOF was found. The printout is:

```
PHYSICAL EOF ON RANDOM ACCESS DEVICE
```

Printout in DUMP Function

Logical Dumps.

The heading line, which is printed once per file, contains the file number and the file code of the file. The contents of the record are then printed with the following information: the Block Number (BLK#), the Record Number (REC#), the lower part of the Record Control Word (RCW(L)), the Word Sequence Number (WRD#) of the first data word of the line, and up to five data words in octal and BCD equivalent. A Record Control Word is, of course, not printed for fixed length records. Refer to Appendix A, Figure A-1, for an example of a Logical Dump.

Physical Record Processing (PHYREC) Dumps

The heading line, which is printed once per file, contains the file number, the file code and the density of the file. The contents of the record are then printed with the following information: the Block Number (BLK#), the Mode of that particular block, the Character Count (CC), the Word Sequence Number (WRD#) of the first data word of the line, and up to five data words in octal and BCD equivalent. Refer to Appendix A, Figures A-2 and A-3, for examples of a PHYREC Dump.

Information Common to both Logical and Physical Dumps

The following information is common to both Physical and Logical Dumps. For the BCD equivalent, the octal 17 (?) and octal 77 (!), special editing characters, are replaced by an octal 37 (/). End-of-file marks are displayed in octal. On a labeled tape, each 14-word label is printed with its file code (if physical, the density and mode are also printed). Duplicate lines are suppressed and an asterisk is shown next to the word number in the line following one or more suppressed lines. Information such as BLK#, REC#, etc, is not printed if it is the same as for the preceding line.

Printout in Random Dump

The heading line, which is printed once per file, contains the file number and the file code of the file. The contents of the record are then printed with the following information: the Record Number (REC#), the Word Sequence Number (WRD#) of the first data word of the line, and up to five data words in octal and BCD equivalent. For the BCD equivalent, the octal 17 (?) and octal 77 (!) special editing characters are replaced by an octal 37 (/). Duplicate lines are suppressed and an asterisk is shown next to the word number in the line following one or more suppressed lines. Refer to Appendix A, Figure A-4, for an example of a Random Dump.

Printout in DDUMP Function

The printout is the same as that described for the DUMP functions, with these exceptions:

1. Only the BCD representation of the data words is shown, up to 14 words per line.
2. An asterisk is printed before and after the record, to show the exact starting and ending positions of the record.

Refer to Appendix A, Figure A-5, A-6 and A-7, for examples of a DDUMP dump.

Printout in ADUMP Function (Time Sharing Files)

The Block Control Word (BCW) is printed before each block. Thereafter, the Record Control Word (RCW) is printed before each record. Contents of the record are printed up to five words per line in octal and ASCII equivalent. The first character, the string count, is replaced by an asterisk (octal 54). All null (unprintable) ASCII characters that transliterate to special editing characters (octal 77 and octal 17) are shown as octal 37 (/) in the ASCII equivalent. Refer to Appendix A, Figure A-8, for an example of a logical ADUMP function dump.

Printout in AADUMP Function (Time Sharing Files)

Each block is preceded by a printout of the Block Control Word (BCW). Thereafter, each string is preceded by the Record Control Word (RCW). The printout is one string per line in ASCII equivalent. A carriage return character (ASCII 015) is replaced by an octal 72 (←) to denote a carriage return. In addition, the string count is converted to a decimal number. All null (unprintable) ASCII characters are replaced by an asterisk (octal 54). ASCII characters that transliterate to special editing characters (octal 77 and octal 17) are shown as octal 37 (/) in the ASCII equivalent. Refer to Appendix A, Figure A-8, for example of a logical AADUMP function.

Printout in ADUMP and AADUMP Using Physical Record Processing

Physical Record Processing (PHYREC) is applicable only to magnetic tape files. The output format is the same as a physical record dump with one difference -- the ASCII equivalent is used instead of the BCD equivalent. Blocks are dumped after each character has been transliterated. All null (unprintable) ASCII characters are replaced by an asterisk (octal 54). ASCII characters that transliterate to special editing characters (octal 77 and octal 17) are shown as octal 37 (/) in the ASCII equivalent. Refer to Appendix A, Figures A-9 and A-10, for examples of ADUMP and AADUMP using PHYREC.

Printout in COPY, MCOPY Function

ILLEGAL END FILE MARK xx--COPIED AS 67.

The xx is the illegal file mark 77.

SECTION IV

DETAILED DESCRIPTION OF UTILITY FUNCTIONS

Utility performs the functions of COPY, MCOPY, COMPare, SKIP, BKSP, DUMP, ADUMP, AADUMP, DDUMP, REWind, HOLD, RPT, RSAVE, RREST, and RCOPY as follows:

COPY

When labeled tapes are copied, input header label words 7-13 of the most recently opened input tape are copied to the output label. Word 7 (creation date) and word 8 (retention days) can be changed by use of the NDATE and RETPER options of the $ FFILE card. Input trailer label words 3-13 are copied to the output trailer label.

MCOPY

Similar to the COPY function with this exception: When a standard octal 17 end of file mark is reached on the input, the output device is not closed but left open for merging multiple files into one single output file. However, the output file is closed if the MCOPY option is followed by a REWind of the output file; or at the end of the $ FUTIL card if there is no HOLD option for the output file.

MCOPY should not be used with physical record processing (PHYREC) if block serial numbers are present (unless the RESEQ option is also used). File and Record Control builds new block serial numbers for output files in the logical record mode, but only copies block serial numbers in the physical record mode. Out of sequence block serial numbers would then be sent to output if MCOPY were used with physical record processing. The RESEQ option on a $ QUTIL card will take care of this problem.

Logical Record Processing

The input file is read, and the record size is transferred from the input record to the output record. The output file is then written to by means of the COPY routine. If both devices call for the standard variable-length record format, the report and media codes are transferred from the input record to the output buffer.

Physical Record Processing

After the input device has been read, the DCW string is adjusted for the correct record length. Writing to the output device from the input buffer is by means of the WRITE routine.

Utility cannot copy or dump checkpoints because the File and Record Control bypasses them and Utility never sees them.

End-of-File Processing (both logical and physical)

If the EOF is not 17 (octal), the file mark is transferred to the output device by Call WEF. If it is a standard 17 (octal) end-of-file mark, the output device is closed. A 23 (octal) filemark is standard on 9-track tape (rather than 17 (octal)).

COMPARE

Record sizes are compared, and if they agree, a word-by-word comparison is made. If an end-of-file-mark is encountered on one device and not in the other, the other device is read (without comparison) until a similar end-of-file occurs. Then, comparison continues. When 50 (or n + 1 as specified in a CMPERR QUTIL parameter) compare errors are encountered, the Utility activity is terminated on a fatal error. Also, even if the upper limit of compare errors is not reached, the existence of one or more compare errors at the completion of the compare function causes a termination of the Utility activity on a fatal error.

SKIP

The first parameter refers to the first file code; the second parameter refers to the second. The device is read until the file/record count (as given in the $ FUTIL card SKIP parameters) is exhausted or a partial label is encountered.

BKSP

The first parameter refers to the first file code; the second parameter to the second. The device is read until the file/record count (as given in the $ FUTIL card BKSP parameters) is exhausted or a beginning-of-tape label is encountered.

DUMP

Information on the device is written out in both octal and BCD formats to P*.

Upon termination of an MCOPY function, the following message is also placed on P*.

TOTAL RECORDS MCOPIED xxxxxx

The record count xxxxxx shows the total number of records MCOPIED to the output.

Example: If the function was MCOPY/3F/ and the three files contained 300, 300, and 302 records respectively, the record count is 902 records.

COMP Function--No Errors. If there were no compare errors and the COMP function was terminated by record countdown, the following message is placed on P*:

COMPARED xxxxxx FILES xxxxxx RECORDS

If there were no compare errors and the COMP function was terminated by an end-of-file having been reached, the following message is placed on P*:

COMPARED xxxxxx FILES. xxxxxx RECORDS IN LAST FILE

COMP Function--With Errors. If there were one or more compare errors when the compare function terminated, one of the following messages is placed on P* and the activity is terminated with a fatal error.

COMP ERROR COUNT xxxxxx

COMPARE COMPLETED. ERROR COUNT xxxxxx

or if the maximum number of compare errors allowed is exceeded, the compare function is aborted and the following message is used:

COMPARE ABORTED. ERROR COUNT xxxxxx

RSAVE, RREST, or RCOPY More or Less Than One File

An attempt to use the RESAVE, RREST, or RCOPY functions for either more or less than one file at one time, results in the following message:

$$\text{UTILITY} \left\{\begin{array}{l}\text{SAVES}\\ \text{RESTORES}\\ \text{RCOPIES}\end{array}\right\} \text{ONLY A FILE}$$

Printout in Compare Function

Every specification of a COMP function results in the following printout that indicates that data on FILE (code) xx is to be compared with data on FILE (code) yy.

COMPARE - xx VS yy

Printout in Backspace Function

Every specification of a BKSP function results in the following printout if the loadpoint is encountered.

```
FILE CODE xx LOADPOINT ON BKSP
```

The following printouts indicate that there was a noncomparison and they identify by number the two files (mm) and two records (nn) which do not compare.

```
FILE # mm -- RECORD # nn, AND FILE # mm -- RECORD # nn

RECORDS DO NOT COMPARE
```

Noncomparison of Data Records. The reason for noncomparison is shown by either of two printouts. The following printout indicates that the two records just compared were not the same size. When sizes do not compare, the COMP ERROR COUNT is incremented by one and no further check is made on the two records. The record sizes are printed, followed by a dump of the two unequal records. The printout is as follows:

```
SIZES DIFFERENT xxxxxx VS yyyyyy
```

where:

xxxxxx and yyyyyy are the sizes of the two records

```
FILE CODE (#1) (DUMP OF FIRST BLOCK)

FILE CODE (#2) (DUMP OF SECOND BLOCK)
```

The following printout indicates that the two compared records are the same size but the data does not compare. When the data does not compare, the following line is printed and the COMP ERROR COUNT is incremented by one for each noncomparison within the two records. The printout is:

```
#### aaaaaaaaaaaa bbbbbbbbbbbb ccccccdddddd
```

where:

#### = the word number within the record where the noncompare occurred
a--a = the octal representation of the data word from file code xx
b--b = the octal representation of the data word from file code yy
c--c = the BCD equivalent of the data word from file code xx
d--d = the BCD equivalent of the data word from file code yy

Noncomparison of End-of-File Marks. The following printouts indicate that the EOF marks are not aligned. They specify which unit first encountered the EOF. The other unit is then aligned from a matching EOF before comparisons are resumed. If the number of files to be compared is exhausted during this positioning, comparison is terminated. The printout is either

```
EOF 1ST UNIT ONLY

        or

EOF 2ND UNIT ONLY
```

When the EOF marks are aligned but do not compare, the second device is positioned until either the file marks agree or the number of files or records to be processed is exhausted. The printout for noncomparing file marks is:

```
END FILE MARK - xx VS yy
```

where:

xx and yy are the disagreeing file marks

Printout for Physical End-of-File

While processing files on a random access device, a physical end-of-file condition may be encountered. Utility treats this as though a standard EOF was found. The printout is:

```
PHYSICAL EOF ON RANDOM ACCESS DEVICE
```

Printout in DUMP Function

Logical Dumps.

The heading line, which is printed once per file, contains the file number and the file code of the file. The contents of the record are then printed with the following information: the Block Number (BLK#), the Record Number (REC#), the lower part of the Record Control Word (RCW(L)), the Word Sequence Number (WRD#) of the first data word of the line, and up to five data words in octal and BCD equivalent. A Record Control Word is, of course, not printed for fixed length records. Refer to Appendix A, Figure A-1, for an example of a Logical Dump.

Physical Record Processing (PHYREC) Dumps

The heading line, which is printed once per file, contains the file number, the file code and the density of the file. The contents of the record are then printed with the following information: the Block Number (BLK#), the Mode of that particular block, the Character Count (CC), the Word Sequence Number (WRD#) of the first data word of the line, and up to five data words in octal and BCD equivalent. Refer to Appendix A, Figures A-2 and A-3, for examples of a PHYREC Dump.

Information Common to both Logical and Physical Dumps

The following information is common to both Physical and Logical Dumps. For the BCD equivalent, the octal 17 (?) and octal 77 (!), special editing characters, are replaced by an octal 37 (/). End-of-file marks are displayed in octal. On a labeled tape, each 14-word label is printed with its file code (if physical, the density and mode are also printed). Duplicate lines are suppressed and an asterisk is shown next to the word number in the line following one or more suppressed lines. Information such as BLK#, REC#, etc, is not printed if it is the same as for the preceding line.

Printout in Random Dump

The heading line, which is printed once per file, contains the file number and the file code of the file. The contents of the record are then printed with the following information: the Record Number (REC#), the Word Sequence Number (WRD#) of the first data word of the line, and up to five data words in octal and BCD equivalent. For the BCD equivalent, the octal 17 (?) and octal 77 (!) special editing characters are replaced by an octal 37 (/). Duplicate lines are suppressed and an asterisk is shown next to the word number in the line following one or more suppressed lines. Refer to Appendix A, Figure A-4, for an example of a Random Dump.

Printout in DDUMP Function

The printout is the same as that described for the DUMP functions, with these exceptions:

1. Only the BCD representation of the data words is shown, up to 14 words per line.
2. An asterisk is printed before and after the record, to show the exact starting and ending positions of the record.

Refer to Appendix A, Figure A-5, A-6 and A-7, for examples of a DDUMP dump.

Printout in ADUMP Function (Time Sharing Files)

The Block Control Word (BCW) is printed before each block. Thereafter, the Record Control Word (RCW) is printed before each record. Contents of the record are printed up to five words per line in octal and ASCII equivalent. The first character, the string count, is replaced by an asterisk (octal 54). All null (unprintable) ASCII characters that transliterate to special editing characters (octal 77 and octal 17) are shown as octal 37 (/) in the ASCII equivalent. Refer to Appendix A, Figure A-8, for an example of a logical ADUMP function dump.

Printout in AADUMP Function (Time Sharing Files)

Each block is preceded by a printout of the Block Control Word (BCW). Thereafter, each string is preceded by the Record Control Word (RCW). The printout is one string per line in ASCII equivalent. A carriage return character (ASCII 015) is replaced by an octal 72 (←) to denote a carriage return. In addition, the string count is converted to a decimal number. All null (unprintable) ASCII characters are replaced by an asterisk (octal 54). ASCII characters that transliterate to special editing characters (octal 77 and octal 17) are shown as octal 37 (/) in the ASCII equivalent. Refer to Appendix A, Figure A-8, for example of a logical AADUMP function.

Printout in ADUMP and AADUMP Using Physical Record Processing

Physical Record Processing (PHYREC) is applicable only to magnetic tape files. The output format is the same as a physical record dump with one difference -- the ASCII equivalent is used instead of the BCD equivalent. Blocks are dumped after each character has been transliterated. All null (unprintable) ASCII characters are replaced by an asterisk (octal 54). ASCII characters that transliterate to special editing characters (octal 77 and octal 17) are shown as octal 37 (/) in the ASCII equivalent. Refer to Appendix A, Figures A-9 and A-10, for examples of ADUMP and AADUMP using PHYREC.

Printout in COPY, MCOPY Function

```
ILLEGAL END FILE MARK xx--COPIED AS 67.
```

The xx is the illegal file mark 77.

SECTION IV

DETAILED DESCRIPTION OF UTILITY FUNCTIONS

Utility performs the functions of COPY, MCOPY, COMPare, SKIP, BKSP, DUMP, ADUMP, AADUMP, DDUMP, REWind, HOLD, RPT, RSAVE, RREST, and RCOPY as follows:

COPY

When labeled tapes are copied, input header label words 7-13 of the most recently opened input tape are copied to the output label. Word 7 (creation date) and word 8 (retention days) can be changed by use of the NDATE and RETPER options of the $ FFILE card. Input trailer label words 3-13 are copied to the output trailer label.

MCOPY

Similar to the COPY function with this exception: When a standard octal 17 end of file mark is reached on the input, the output device is not closed but left open for merging multiple files into one single output file. However, the output file is closed if the MCOPY option is followed by a REWind of the output file; or at the end of the $ FUTIL card if there is no HOLD option for the output file.

MCOPY should not be used with physical record processing (PHYREC) if block serial numbers are present (unless the RESEQ option is also used). File and Record Control builds new block serial numbers for output files in the logical record mode, but only copies block serial numbers in the physical record mode. Out of sequence block serial numbers would then be sent to output if MCOPY were used with physical record processing. The RESEQ option on a $ QUTIL card will take care of this problem.

Logical Record Processing

The input file is read, and the record size is transferred from the input record to the output record. The output file is then written to by means of the COPY routine. If both devices call for the standard variable-length record format, the report and media codes are transferred from the input record to the output buffer.

Physical Record Processing

After the input device has been read, the DCW string is adjusted for the correct record length. Writing to the output device from the input buffer is by means of the WRITE routine.

Utility cannot copy or dump checkpoints because the File and Record Control bypasses them and Utility never sees them.

End-of-File Processing (both logical and physical)

If the EOF is not 17 (octal), the file mark is transferred to the output device by Call WEF. If it is a standard 17 (octal) end-of-file mark, the output device is closed. A 23 (octal) filemark is standard on 9-track tape (rather than 17 (octal)).

COMPARE

Record sizes are compared, and if they agree, a word-by-word comparison is made. If an end-of-file-mark is encountered on one device and not in the other, the other device is read (without comparison) until a similar end-of-file occurs. Then, comparison continues. When 50 (or n + 1 as specified in a CMPERR QUTIL parameter) compare errors are encountered, the Utility activity is terminated on a fatal error. Also, even if the upper limit of compare errors is not reached, the existence of one or more compare errors at the completion of the compare function causes a termination of the Utility activity on a fatal error.

SKIP

The first parameter refers to the first file code; the second parameter refers to the second. The device is read until the file/record count (as given in the $ FUTIL card SKIP parameters) is exhausted or a partial label is encountered.

BKSP

The first parameter refers to the first file code; the second parameter to the second. The device is read until the file/record count (as given in the $ FUTIL card BKSP parameters) is exhausted or a beginning-of-tape label is encountered.

DUMP

Information on the device is written out in both octal and BCD formats to P*.

ADUMP

Information on the device is written out in both octal and ASCII formats to P*.

AADUMP

Information on the device is written out only in ASCII format to P*.

DDUMP

Information on the device is written out only in BCD format to P*.

REWIND

The appropriate device is rewound.

HOLD

The file control block pertaining to this particular file code is not closed at the termination of this $ FUTIL card's functions. The HOLD function is usually needed only with a $ FUTIL card on that the previous Utility function performed has specified records not files. This is because a file is closed by Utility at the completion of a function that specifies a file(s) (except for the MCOPY function). Consequently, a HOLD operation cannot hold open a file which Utility has already closed. One exception is the MCOPY function. HOLD can be used after an MCOPY function because use of the MCOPY function does not have closed the output file. See MCOPY, as previously described in this section.

It may be necessary to use HOLD when reading multiple files from a random access device. At the end of all processing for a FUTIL card, not only are the files being used closed, but also their file control blocks are released. New file control blocks would be generated for additional FUTIL cards. This must not happen if the next file(s) on the random access device is to be read correctly. The HOLD option must be used in this case to keep the input file control block from being released between FUTIL cards.

RPT

Repeat the next m functions n times. The RPT function plus the repeat functions must be contained on one $ ETC control card that continues a $ FUTIL control card. Nesting of a RPT function within itself is not permitted--you cannot repeat a repeat function. If a partial label is encountered during RPT processing, all processing terminates. A partial label is a label written by File and Record Control as an end of data banner on all magnetic tapes.

RSAVE

1281-word physical records (1280 words plus the Block Serial Word) are written to a labeled magnetic tape. The last physical record written can contain less than 1280 words; i.e., it can contain 320, 640, or 960 words plus the Block Serial Word.

RREST

Utility reads a magnetic tape formatted as described in RSAVE above and writes it to a random file. Block serial numbers are checked as the tape is read. If a block serial error is encountered, the rest of the activity is aborted with an appropriate message. Multiple files can be saved on a magnetic tape by means of multiple RSAVE functions and any or all of these files can be restored by means of RREST functions.

RCOPY

Utility copies by 1280-word blocks (when possible) from the random input file to the random output file. The last block copied can be 320, 640, or 960 words if the block is less than 1280 words.

SECTION V

OWN CODE OPERATIONS WITH UTILITY

OWN CODE CAPABILITY

OWN CODE capability is a Utility feature that permits the user to change his files without having to write a separate program each time file maintenance is required.

The user, by selecting the appropriate OWN CODE option, can specify that user coding be given control at any one of five points during a Utility run. The OWN CODE options (options 2-5 are effective only during a Utility COPY or MCOPY function) are as follows:

1. INITIALIZATION: If this option is selected, entry is made only once to OWN CODE for the option during Utility initialization procedures.

2. COPY or MCOPY: In this option, entry is made to OWN CODE immediately after each input record has been read, but before the record has been copied.

3. END OF FILE: If this option is selected, entry is made to OWN CODE each time File and Record Control makes an end-of-file return to Utility.

4. PRELABEL (input): In this option, entry to OWN CODE is made immediately after each standard input tape label has been read (but not yet checked). This is equivalent to using File and Record Controls prelabel entries.

5. POSTLABEL (output): This option is similar to the PRELABEL option and is equivalent to File and Record Control's post label entries. In this POSTLABEL option, entry to OWN CODE is made immediately after each magnetic tape label has been built (but before it has been written).

An OWN CODE link (previously placed on H* by the user) is called by Utility if the first Utility control card in the activity is $ QUTIL USER. Note that, a $ LIMITS control card could precede the $ QUTIL USER card, because the $ LIMITS card is not a Utility control card.

OWN CODE FORMAT

The OWN CODE link on H* must contain an option table as well as the user's routines for the options specified in the table. The user specifies in the option table the option(s) needed. The option table must be six words long.

The following table shows the option table format:

TABLE	INIT Option Entry	MBZ	Word 1
+1	COPY or MCOPY Option Entry	MBZ	Word 2
+2	EOF Option Entry	MBZ	Word 3
+3	PRELABEL Option Entry	MBZ	Word 4
+4	POSTLABEL Option Entry	MBZ	Word 5
+5	FCB1	FCB2	Word 6

An option is effectively selected if the appropriate position in the table contains the starting address of the user's coding for this option. If an option is not used, its position in the table must contain zeros. Word 6 of the table is filled by Utility with the addresses of Utility's two file control blocks.

USING OWN CODE

OWN CODE must be present as a link on H*. The presence of a $ QUTIL USER card as the first Utility control card causes Utility to read in the OWN CODE link from H* and to set up linkages between OWN CODE and itself.

The $ QUTIL USER card must be set up as follows:

1	8	16
$	QUTIL	USER

No other parameters are permitted on this card. If other QUTIL options are required for this activity, they must be specified on other QUTIL cards. The USER parameter on a $ QUTIL card is legal only if the $ QUTIL card is the first Utility control card for the Utility activity.

The following is a sample deck setup showing the use of OWN CODE:

```
            1         8              16
           ________________________________________________
 1          $         LOWLOAD        8956
 2          $         OPTION         NOGO,NOSETU,SAVE/USER
                       . }
                       . } OWN CODE (either source deck for
                       . } assembly or an object deck)

            $         EXECUTE        DUMP
 3          $         TAPE           H*,X7S
            $         UTILITY
            $         LIMITS         ,16000
 3          $         TAPE           H*,X7R
            $         QUTIL          USER
                       . }
                       . } Other Utility control cards
                       . }

            $         ENDJOB
```

The numbers in this sample listing refer to the following definitions:

1. The OWN CODE link must be loaded above UTILITY.

2. The name of the link containing OWN CODE must be USER.

3. The logical unit designator for H* may be any logical unit designator not being used at this time.

APPLYING OWN CODE OPTIONS

The following points pertain to all OWN CODE options:

1. Transportation between Utility and OWN CODE is via index register 1 (X1), and OWN CODE must preserve the contents of X1 in order to make a correct return to Utility.

2. All registers except X1 are free for use by OWN CODE. Utility safestores the contents of its registers and indicators before transferring control to OWN CODE and restores them immediately after return to Utility.

3. Except for the INITIALIZATION option, OWN CODE is given control only during the COPY or MCOPY function. For example, OWN CODE for the PRELABEL option does not receive control during a DUMP, SKIP or COMPARE function. In the INITIALIZATION option, OWN CODE gets control only once, and that is during Utility's initialization procedures. OWN CODE can be used indirectly for the DUMP function by having OWN CODE make a new file with the COPY or MCOPY function and then doing a DUMP or DDUMP of the new file.

4. Utility fills word 6 of the option table with the addresses of Utility's two file control blocks.

FILE CONTROL BLOCKS

The address of the input file control block (FCB1) is stored in the left half of word 6 of the option table. The address of the output file control block (FCB2) is stored in the right half of word 6. The user may interrogate these blocks in his OWN CODE, but must not change them in any way.

There is a special point to be considered in the interrogation of the input file control block. File and Record Control increases the block count in LOCSYM-6 in FCB1 for each nonstandard file mark read. This is not increased in FCB2 for each nonstandard file mark written.

No attempt should be made to interrogate the file control blocks in OWN CODE for the INITIALIZATION option, because the file control blocks have not yet been set up.

OWN CODE STORAGE REQUIREMENTS

When read in from H*, OWN CODE is loaded at the beginning (Utility is lowloaded) of Utility's available storage area.

This area is normally used by Utility for I/O buffers. After OWN CODE is loaded, Utility adjusts its "beginning of available storage" pointers so that the buffers are setup immediately above the OWN CODE. All OWN CODE processing increase Utility's memory requirement to more than its standard allocation of 10K; therefore a $ LIMITS card requesting more than 10K of memory is required. The load map for the OWN CODE link show the actual storage needed for the OWN CODE.

EXAMPLES OF OWN CODE OPTIONS

Each of the OWN CODE options is defined in the following examples. Each example illustrates the use of one option; however, any combination of options (including use of all options) can be used in one activity.

INITIALIZATION Option

If the OWN CODE entry location for this option is specified in the option table, OWN CODE is entered once during Utility's initialization. No attempt should be made to interrogate Utility's file control blocks in the OWN CODE for this option, since the file control blocks have not yet been set up.

Example

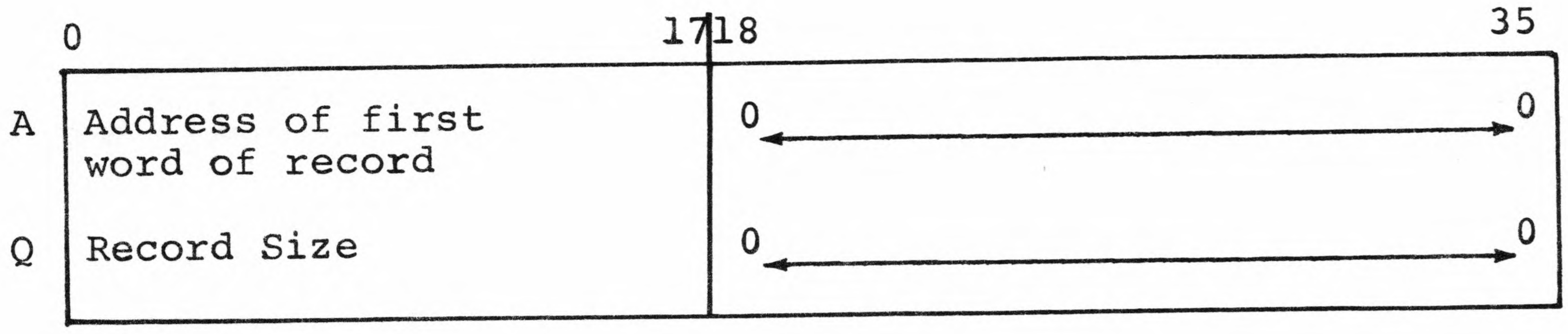
OWN CODE Usage Identification. In the following example of OWN CODE, a one line message is put on SYSOUT before Utility begins its processing:

```
$       GMAP
        SYMDEF    TABLE
TABLE   ARG       INIT
        ARG       0                          COPY
        ARG       0                          EOF
        ARG       0                          PRE
        ARG       0                          POST
FCBS    ZERO      **,**                      UTILITY'S FCB'S
INIT    STX1      EXIT                       SAVE EXIT ADDRESS
        CALL      OPEN(LIST,1)               OPEN SYSOUT FILE
        CALL      PRINT(FCB,MESS,SLEW2)      PRT MESSAGE
        CALL      CLOSE(LIST,1)              CLOSE SYSOUT FILE
EXIT    TRA       **
LIST    VFD       18/FCB,1/1
        FILCB     FCB,AB,BUF,,,,6
MESS    BCI       6,THIS IS AN OWN CODE PASS OF UTILITY
SLEW2   DEC       2
BUF     BSS       321
        END
```

Among the control cards for this Utility activity using OWN CODE must be a $ SYSOUT AB file card for file AB.

COPY/MCOPY Option

When the COPY or MCOPY option is specified, control is turned over to OWN CODE immediately after each input record has been fetched, but before it has been copied. When control is turned over to the OWN CODE, the A- and Q-registers contain the following:



If logical record processing has been specified, the address in the high order half of the A-register is the current record index. In physical record processing, the address is that of the first word of the tape block.

At this time, index register 2 (X2) contains the address of Utility's input file control block. In this way, if the address is needed, it does not have to be obtained from word 6 of the option table.

Returns to Utility. There are two returns to Utility from OWN CODE:

1. TRA 0,1

   If this return is used, Utility gets the next input record.

2.
```
L      TSX6      1,1

L+1    Zero      X,0

L+2    Zero      Y,0

L+3    Normal Return
```

   X = Address of first word of record to be written

   Y = Record size

   If this return is used, Utility writes to the output file the record whose starting address is specified by C(L+1)0-17 and whose length is specified by C(L+3)0-17. It then returns to L+3 of the OWN CODE.

Example 1 -

Record Insertion. If the second return is used to insert a variable length record from core storage, the user should be aware that Utility now considers L+1 as the current record index of the input record. Utility assumes that the report and media code word for the record is located at "address-1," relative to the address given by the current record index. Therefore, the user must supply the report and media code, right justified, in a word in core storage preceding the first data word of that record.

In the following example, logical record processing is assumed and a new record is inserted into the file. In addition, a new report code (octal 66) and media code (octal 3) is copied from a word preceding the first data word in storage.

```
$         GMAP
          SYMDEF      TABLE
TABLE     ARG         0                        INIT
          ARG         COPY
          ARG         0                        EOF
          ARG         0                        PRE
          ARG         0                        POST
FCBS      ZERO        **,**                    UTILITY FILE CONTROL BLOCKS
COPY      STAQ        SAVAQ                    SAVE LOC AND LENGTH
          AOS         COUNT
          LDA         COUNT
          CMPA        =6,DL                    IS THIS SIXTH RECORD
          TNZ         WRITE                    NO
          TSX6        1,1                      YES INSERT NEW RECORD
          ARG         NEWREC                   LOC OF NEW RECORD
          ARG         5                        LENGTH
WRITE    OTSX6        1,1                      COPY RECORD
SAVAQ     DEC         0                        LOC OF RECORD
          DEC         0                        LENGTH
          TRA         0,1                      RETURN
MEDIA     OCT         366                      REPORT AND MEDIA CODE FOR
                                                NEW RECORD
NEWREC    BCI         5,THIS IS A NEW RECORD 7/7/69
COUNT     DEC         0
          END
```

Example 2 -

Changing One Word of a Record. The following example is used to change the block count in the first word of a physical record. This count is normally incremented by File and Record Control and is reset to one for the next reel in a multireel file. This example uses both the COPY and POST options and assumes physical record processing. Each time a header label is recognized, OWN CODE reinitializes the block count to zero. It also keeps its own block count for rebuilding block serial words on the output.

```
$         GMAP
          SYMDEF      TABLE
TABLE     ARG         0                        INIT
          ARG         COPY
          ARG         0                        EOF
          ARG         0                        PRE
          ARG         POST
FCBS      ZERO        **,**                    UTILITY FCB'S
POST      LDAQ        0,2
          CMPAQ       HLBL                     IS THIS A HEADER LABEL
          TNZ         0,1                      NO, RETURN
          STZ         COUNT                    YES, ZERO COUNT
          TRA         0,1                      RETURN
COPY      STAQ        SAVAQ                    SAVE WORD LOC AND REC SIZE
          AOS         COUNT                    ADD 1 TO COUNT
          LXL3        COUNT                    STORE COUNT IN BLOCK SERIAL
                                                 WORD
          STX3        0,AU                     IN AU
WRITE    OTSX6        1,1                      WRITE RECORD
SAVAQ     DEC         0
          DEC         0
          TRA         0,1
COUNT     DEC         0
HLBL      EBCI        2,GEb̸b̸600b̸BTL
          END
```

The above example is intended to be only illustrative in nature. The user would undoubtedly use the RESEQ option instead to reserialize block serial numbers.

Example 3 -

Changing and Inserting Words in a Record. In the following example of the COPY option, physical record processing is assumed. This processing does the following:

1. Places all zeros in word 2 of block 3.
2. Inserts a new block (five words) from storage between blocks 5 and 6.
3. Starting with block 7, doesn't copy blocks which have a third word of all zeros.

By proper use of the two methods of returning to Utility, a user can easily delete, modify or insert records.

```
$        GMAP
         SYMDEF     TABLE
TABLE    ARG        0                       INIT
         ARG        COPY
         ARG        0                       EOF
         ARG        0                       PRE
         ARG        0                       POST
FCBS     ZERO       **,**                   UTILITIES FCB'S
COPY     STAQ       SAVAQ                   SAVE LOC AND LENGTH
         LDQ        -6,2                    GET
         ANQ        =O777777,DU             BLOCK NUMBER
         CMPQ       =3,DU                   THIRD BLOCK
         TZE        THIRD                   YES
         CMPQ       =6,DU                   IS THIS 6th BLOCK
         TZE        SIXTH                   YES
         SZN        2,AU                    IS THIRD WORD ALL ZERO
         TZE        0,1                     YES, DON'T COPY
WRITE   OTSX6       1,1                     COPY RECORD
SAVAQ    DEC        0                       LOC OF RECORD
         DEC        0                       RECORD LENGTH
         TRA        0,1                     GET ANOTHER RECORD
THIRD    STZ        1,AU
         TRA        WRITE
SIXTH    TSX6       1,1                     INSERT NEW BLOCK
         ARG        NEWBLK
         ARG        5
         TRA        WRITE                   COPY SIXTH BLOCK
NEWBLK   DEC        1,2,3,4,5
         END
```

If the input tape had block serial numbers, either OWN CODE would have had to resequence the block serial numbers or the RESEQ option would have had to be used.

Example 4 -

Creating Test Files From One Logical Record. In this example, a test file of 25 three-word logical records is created from one logical record that could be input to Utility on a $ DATA file. The 24 additional records differ from the original record only in the second word, incremented by one for each succeeding record.

```
$         GMAP
          SYMDEF        TABLE
TABLE     ARG           0                                INIT
          ARG           COPY
          ARG           0                                EOF
          ARG           0                                PRE
          ARG           0                                POST
FCBS      ZERO          **,**                            UTILITY FCB'S
COPY      STA           SAVA                             ADDRESS OF FIRST WORD OF REC
TEST      LDA           RECCNT
          CMPA          =25,DL                           HAVE WE WRITTEN 25 RECORDS
          TZE           0,1                              YES RETURN
          TSX6          1,1
SAVA      DEC           0
          ARG           3
          LDX2          FCBS,I                           LOAD WITH CRX
          AOS           1,2
          AOS           RECCNT                           ADD TO REC CNT
          TRA           TEST
RECCNT    DEC           0
          END
```

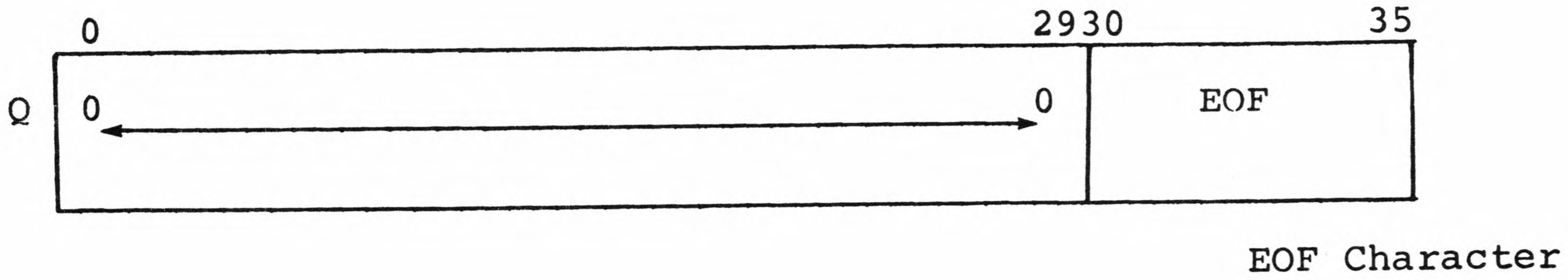
The FUTIL card for the activity could be as follows:

```
1         8           16
______________________________________
$         FUTIL       F1,F2,RWD/F1,F2/,COPY/1R/,RWD/F2/
```

END-OF-FILE Option

In the end-of-file (EOF) option, OWN CODE is given control each time an EOF status is returned to Utility. When control is turned over to OWN CODE, the Q-register contains the EOF character that was returned to Utility, OWN CODE should return to Utility as follows:

```
TRA       0,1
```



A standard EOF on nine-track tape (23 octal) is an exception. The Q-register contains a 17 (octal) when a standard EOF is encountered on nine-track tape.

Depending on the EOF in the Q-register, bits 30-35, Utility takes the following action:

EOF	Utility Action
00	Do not put an EOF on output file.
17 (octal)	Close output file. (Utility puts out standard EOF's by closing output file.)
77 (octal)	Take standard action as if OWN CODE was not used.
All others	Write nonstandard EOF in Q lower to output file.

Example 1 -

Changing EOF Marks. As indicated in the table, EOFs can be changed or deleted easily with the OWN CODE end-of-file option. If EOFs are to be added, they must be added by using the COPY option.

```
$       GMAP
        SYMDEF    TABLE
TABLE   ARG       0                      INIT
        ARG       0                      COPY
        ARG       EOF
        ARG       0                      PRE
        ARG       0                      POST
FCBS    ZERO      **,**                  UTILITY FCB'S
EOF     CMPQ      =O16,DL                IS IT A O16 EOF
        TZE       CHANGE
RET     TRA       0,1                    NO, RETURN SAME EOF
CHANGE  LDQ       =O15,DL                CHANGE TO O15
        TRA       RET                    AND RETURN TO UTILITY
        END
```

In this example, any 16 (octal) EOFs encountered on the input file(s), are copied as 15 (octal) EOFs.

PRELABEL Processing Option

In the PRELABEL processing option, OWN CODE receives control immediately after an input magnetic tape label has been read but before it is checked. Index register 2 (X2) contains the starting location (always an even numbered location) of the label. Return to Utility must be by a TRA 0,1. The following File and Record Control label exit restriction must be observed: user code must not issue a logical record request for the file involved in the label procedure.

Example -

Verifying File Count on Tape. The following is an example of the PRELABEL processing option:

```
$       GMAP
        SYMDEF    TABLE
TABLE   ARG       0                      INIT
        ARG       0                      COPY
        ARG       0                      EOF
        ARG       PRE
        ARG       0                      POST
FCBS    ZERO      **,**                  UTILITY FCB'S
PRE     LDAQ      0,2
        CMPAQ     HLBL                   IS THIS HEADER LABEL
        TNZ       0,1                    NO, RETURN
        RPT       6,1,TNZ                CHECK FOR PARTIAL LABEL
        LDA       4,2                    RECOGNIZED BY ZEROS IN
        TZE       *+2                    WORDS 5 THROUGH 10
        TRA       0,1
        LDX3      FCBS                   GET INPUT FOR ADDRESS
        LDQ       -6,3                   GET FILE COUNT
        ANQ       =O77,DL                AND ISOLATE
TEST    CMPQ      =5,DL                  HAVE AT LEAST 4 DATA FILES
                                          BEEN READ
        TRC       0,1                    YES, RETURN
        SBQ       =1,DL                  SUBTRACT FOR PARTIAL LABEL
        MME       GEBORT                 ABORT WITH FILE COUNT FOR REASON
                                          CODE

HLBL   EBCI       2,GEƀƀ600ƀBTL
```

This OWN CODE example verifies that there are at least four data files on a magnetic tape. If there are less than four files on the tape, the job is aborted with the actual file count appearing as the abort reason code. The partial label is counted as the fifth file.

POSTLABEL Processing Option

In the POSTLABEL processing option, OWN CODE receives control immediately after each output magnetic tape label has been built, but before it is written. Index register 2 (X2) contains the starting location (always an even numbered location) of the label. Return to Utility must be by a TRA 0,1. The following File and Record Control label exit restriction must be observed: user code must not issue a logical record request for the file involved in the label procedure.

Example

Changing EOF to EOR. In the following example of the POSTLABEL processing option, an EOF trailer label is changed to an EOR trailer label:

```
$      GMAP
       SYMDEF   TABLE
TABLE  ARG      0                    INIT
       ARG      0                    COPY
       ARG      0                    EOF
       ARG      0                    PRE
       ARG      POST
FCBS   ZERO     **,**                UTILITY FCB'S
POST   LDA      0,2
       CMPA     EOFLB                IS IT AN EOF LABEL
       TNZ      0,1                  NO, RETURN
       LDA      EORLB                CHANGE
       STA      0,2                  TO EOR LABEL
       TRA      0,1                  RETURN
EORLB  BCI      1,ƀEOR
EOFLB  BCI      1,ƀEOF
       END
```

APPENDIX A

EXAMPLES OF DUMP FUNCTIONS

This appendix illustrates examples of various dump functions available to the user of Utility.

```
$       FUTIL    01,,REW/01/,DUMP/20R/
                                    TAPE LABEL FILECODE - 01

                                1  272520200600 002022634320 272520473067 200204060011 200204060011      GE  600 BTL GE PHX 24609 24609
                                6  202000000001 200703010002 202020000000 202020202020 202020202020        0001 73102   000
                               11  202020202020 202020202020 202020202020 202020202020

BLK#  REC#      RCW(L) WRD#                           LOGICAL DUMP FILE#     1 FILECODE 01

   1     1      000321    1  260120512547 465163202120 000000000001 770300000000                  F1 REPORT A 000001\30000

         2      000322    1  260120512547 465163202220 000000000001 770200000000                  F1 REPORT B 000001\20000

         3      000323    1  260120512547 465163202320 000000000001 770100000000                  F1 REPORT C 000001\10000

         4      000321    1  260120512547 465163202120 000000000002 770300000000                  F1 REPORT A 000002\30000

   2     5      000322    1  260120512547 465163202220 000000000002 770200000000                  F1 REPORT B 000002\20000

         6      000323    1  260120512547 465163202320 000000000002 770100000000                  F1 REPORT C 000002\10000

         7      000321    1  260120512547 465163202120 000000000003 770300000000                  F1 REPORT A 000003\30000

         8      000322    1  260120512547 465163202220 000000000003 770200000000                  F1 REPORT B 000003\20000

   3     9      000323    1  260120512547 465163202320 000000000003 770100000000                  F1 REPORT C 000003\10000

        10      000321    1  260120512547 465163202120 000000000004 770300000000                  F1 REPORT A 000004\30000

        11      000322    1  260120512547 465163202220 000000000004 770200000000                  F1 REPORT B 000004\20000

        12      000323    1  260120512547 465163202320 000000000004 770100000000                  F1 REPORT C 000004\10000

   4    13      000321    1  260120512547 465163202120 000000000005 770300000000                  F1 REPORT A 000005\30000

        14      000322    1  260120512547 465163202220 000000000005 770200000000                  F1 REPORT B 000005\20000

        15      000323    1  260120512547 465163202320 000000000005 770100000000                  F1 REPORT C 000005\10000

        16      000321    1  260120512547 465163202120 000000000006 770300000000                  F1 REPORT A 000006\30000

   5    17      000322    1  260120512547 465163202220 000000000006 770200000000                  F1 REPORT B 000006\20000

        18      000323    1  260120512547 465163202320 000000000006 770100000000                  F1 REPORT C 000006\10000

        19      000321    1  260120512547 465163202120 000000000007 770300000000                  F1 REPORT A 000007\30000

        20      000322    1  260120512547 465163202220 000000000007 770200000000                  F1 REPORT B 000007\20000
```

Figure A-1. Logical Dump

```
$      FFILE   01,PHYREC
$      FUTIL   01,,REW/01/,DUMP/5R/
                            TAPE LABEL FILECODE - 01                    DENS  HIGH  MODE BINARY

                          1  272520200600 002022634320 272520473067 200204060011 200204060011      GE  600 BTL GE PHX 24609 24609
                          6  202000000001 200703010002 202020000000 202020202020 202020202020        0001 73102   000
                         11  202020202020 202020202020 202020202020 202020202020

 BLK#       MODE    CC    WRD#                          PHYREC  DUMP FILE#     1 FILECODE 01              DENS  HIGH

   1        BINARY           1  000001000024 000004000321 260120512547 465163202120 000000000001      00100D00403AF1 REPORT A 000001
                             6  770300000000 000004000322 260120512547 465163202220 000000000001      \30000000403BF1 REPORT B 000001
                            11  770200000000 000004000323 260120512547 465163202320 000000000001      \20000000403CF1 REPORT C 000001
                            16  770100000000 000004000321 260120512547 465163202120 000000000002      \10000000403AF1 REPORT A 000002
                            21  770300000000                                                          \30000

   2                         1  000002000024 000004000322 260120512547 465163202220 000000000002      00200D00403BF1 REPORT B 000002
                             6  770200000000 000004000323 260120512547 465163202320 000000000002      \20000000403CF1 REPORT C 000002
                            11  770100000000 000004000321 260120512547 465163202120 000000000003      \10000000403AF1 REPORT A 000003
                            16  770300000000 000004000322 260120512547 465163202220 000000000003      \30000000403BF1 REPORT B 000003
                            21  770200000000                                                          \20000

   3                         1  000003000024 000004000323 260120512547 465163202320 000000000003      00300D00403CF1 REPORT C 000003
                             6  770100000000 000004000321 260120512547 465163202120 000000000004      \10000000403AF1 REPORT A 000004
                            11  770300000000 000004000322 260120512547 465163202220 000000000004      \30000000403BF1 REPORT B 000004
                            16  770200000000 000004000323 260120512547 465163202320 000000000004      \20000000403CF1 REPORT C 000004
                            21  770100000000                                                          \10000

   4                         1  000004000024 000004000321 260120512547 465163202120 000000000005      00400D00403AF1 REPORT A 000005
                             6  770300000000 000004000322 260120512547 465163202220 000000000005      \30000000403BF1 REPORT B 000005
                            11  770200000000 000004000323 260120512547 465163202320 000000000005      \20000000403CF1 REPORT C 000005
                            16  770100000000 000004000321 260120512547 465163202120 000000000006      \10000000403AF1 REPORT A 000006
                            21  770300000000                                                          \30000

   5                         1  000005000024 000004000322 260120512547 465163202220 000000000006      00500D00403BF1 REPORT B 000006
                             6  770200000000 000004000323 260120512547 465163202320 000000000006      \20000000403CF1 REPORT C 000006
                            11  770100000000 000004000321 260120512547 465163202120 000000000007      \10000000403AF1 REPORT A 000007
                            16  770300000000 000004000322 260120512547 465163202220 000000000007      \30000000403BF1 REPORT B 000007
                            21  770200000000                                                          \20000
```

Figure A-2. PHYREC Dump

```
$       FFILE    01,PHYREC
$       FUTIL    01,,REW/01/,DUMP/4R/
  BLK#          MODE    CC    WRD#                                PHYREC  DUMP FILE#     1 FILECODE 01            DENS  HIGH

    1           BCD     4       1  752203030375 330000057505 111120202020 202020202020 202020202020    =B333=.005=599
                                6  202020202020 202020202020 202020202020 202020202020 202020202020                              =
                               11  202020202020 202020202020 202020202020 202020202020 202020202075    00300=00000=00000
                               16  000000000075 000000000075 000000000020 202020202020 202020202020
                               21  202020202020 202020202020 202020202020 202020202020 202020202020         =B407L=,835=/I99=H345=
                               56* 202020202020 202075220400 074375731003 057561311111 753003040575    C345787=B342/=,835=M%U1814R=C8
                               61  230304050710 077522030402 617573100305 754474640110 010451752310    33802=B362/=,835=M%U1834R=B418
                               66  030310000275 220306026175 731003057544 746401100304 517522040110    K=B362=.000199=B407=B521F=,835
                               71  427522030602 753300000001 111175220400 077522050201 267573100305    =L835I98=M805658=B4768074=,I98
                               76  754310030531 111075441000 050605107522 040706100007 047573311110    =L%U1W50R=B637=A806807=B441=L%
                               81  754374640166 050051752206 030775211000 061000077522 040401754374    U1W50R=B637=B676L=]835=/W49=H5
                               86  640166050051 752206030775 220607064375 341003057561 660411753005    00=C500011=B497/=BZ00=M810658=
                               91  000075230500 000001017522 041107617522 710000754410 010006051075    ]835=B558B=L%U1^00R=B637=B676L
                               96  341003057522 050510227543 746401400000 517522060307 752206070643    =B^00=L%U1^00R=B637=,835=L835I
                              101  752240000075 437464014000 005175220603 077573100305 754310030531    98=L%U1^00R=B637=L%U1^00R=B637
                              106  111075437464 014000005175 220603077543 746401400000 517522060307    =,I98=L%U1^00R=B637=B676L=]835
                              111  757331111075 437464014000 005175220603 077522060706 437534100305    I98=B^00=H675=H089=Y811089=BIY
                              116  311110752240 000075300607 057530001011 757010010100 101175223170    8000\=B=B=B=B=B=B=B=B=B=B=B=
                              121  100000003775 227522752275 227522752275 227522752275 227522752275    B000=H089=H726=Y811089=MIZ0717
                              126  220000007530 001011753007 020675701001 010010117544 317100070107    =U%U1B=M811812=L%U1000R=B727L=
                              131  756474640122 754410010110 010275437464 010000005175 220702074375    B000=U%U1B=S806812=V710812B=.0
                              136  220000007564 746401227562 100006100102 756507010010 010222753300    00191=M717768=L%U1000R=B698E=.
                              141  000001110175 440701070706 107543746401 000000517522 061110257533    000111=B723=U99=999999    HEAD
                              146  000000010101 752207020375 641111751111 111111112020 202030252124    R=W63=A= =^13=I= =
                              151  517566060375 217520754001 037531752075 202020202020 202020202020             00
                              156  202020202020 202020200000

    2                   2       1  202020202021 632523304320 202020302521 245120202020 202020202020         ATECHL    HEADR
                                6  202020202020 202020202020 202020202020 202020202020 202020202020
                               11  202020202020 202020202020 202020000704 200100000000                                074 10000

    3                   2       1  202020202021 632567670020 202020242366 202014001420 202020202020         ATEXX0    DCW  @@
                                6  202020202020 202020202346 456263214563 525200202020 202020202020               CONSTANT--0
                               11  202020202020 202020202020 202020000704 200100000000                                074 10000

    4                   2       1  202020202020 202020202020 202020242366 202014221420 202020202020                   DCW  @@
                                6  202020202020 202020202346 456263214563 525247436462 200220202020               CONSTANT--PLUS 2
                               11  202020202020 202020202020 202020000704 200100000000                                074 10000
```

Figure A-3. PHYREC Dump (With Non-Mod 6 Type Records)

```
$      FFILE   F1,RANDOM/0
$      FUTIL   F1,,REW/F1/,DUMP/6R/
        REC#              WRD#                              RANDOM  DUMP FILE#     1 FILECODE F1

         1                  1  000001000477 000024000500 000001000000 000000000012 000000000000      00104\00D0500010000000000[000000
                            6  000000000000 000000000000 000000000000 000000000000 000000000000      000000000000000000000000000000
                           21* 000000000000 000000000000 000450000500 046060060061 060101114114      00000000000004Q0504++60/6119J@
                           26  040124110101 124040107114 111124124105 122123040111 123040116117      41D911[^^8Z@99D[J5[AC419[H^9/\
                           31  124040107114 111124124105 122056015051 060060062060 101040121125      [^^8Z@99D[J5[ ;1QR60+6 +88^[9E
                           36  111103113040 106117130040 112125115120 123040117126 105122040124      9939H^8/\#0^9AE9R [H^9ZF8RB41D
                           41  110105040114 101132131040 104117107056 015110060060 063060116117      91541@89&#8^8J\8Y;1R860+6H+9/\
                           46  127040111123 040124110105 040124111115 105040106117 122040101114      [Y^99C41D91541D99:8Q^8/\[ ^89@
                           51  114040107117 117104040115 105116040124 117040103117 115105040124      9^^8Z\9Z441:8R>41D9Y^8I\9R541D
                           56  117040124110 105040101111 104040117106 040124110105 111122040120      9Y^[J88Q^8998^^9Z641D91599B41
                           61  101122124131 056015033060 060064060120 101125114040                   89B[JI5+:3H+60U61 89E9^^

         2                  1  111123040111 116040114111 113105040106 114131116116 056015115060      99C4199+^9J99I54169JI9/>5+:9Q+
                            6  060065060117 116105040117 106040124110 105040115101 111116040103      60V61\9/541\8+^[J88Q^9R199>413
                           11  101125123105 123040117106 040124105105 116055101107 105040101103      89E[I5[H^9Z641D8R59+)8978Q^893
                           16  103111104105 116124123040 111123040110 125107107111 116107040124      8I98J59/D[H^99C418[R78Z99/741D
                           21  110105040127 122117116107 040103125122 126105056015 051060060066      91541G[A\9/7413[RB[/55+:58+60W
                           26  060124110111 123040111123 040124110105 040105116104 040117106040      61D919[H^99C41D9154159/441\8+^
                           31  124110105040 106125124111 114111124131 040124105123 124015000036      [J88Q^8/E[J99J9[JI41D8RC[^:00<
                           36  000000000000 000000000000 000000000000 000000000000 000000000000      000000000000000000000000000000
                           61* 000000000000 000000000000 000000000000 000000000000                   000000000000000000000000

         3                  1  000000000000 000000000000 000000000000 000000000000 000000000000      000000000000000000000000000000
                           61* 000000000000 000000000000 000000000000 000000000000                   000000000000000000000000

         4                  1  000000000000 000000000000 000000000000 000000000000 000000000000      000000000000000000000000000000
                           61* 000000000000 000000000000 000000000000 000000000000                   000000000000000000000000

         5                  1  000000000000 000000000000 000000000000 000000000000 000000000000      000000000000000000000000000000
                           61* 000000000000 000000000000 000000000000 000000170000                   0000000000000000000000\00

         6                  1  000024000475 000006400600 067065065060 072101104123 103071072115      00D04=006^606YV6Q+7A18JC8HZ7A:
                            6  111103116125 115054061054 061060177177 000006600600 067065066060      9939/E9Q*68*68+\Z\006+606YV6++
                           11  072101104123 103066072111 116102125106 053061065054 065054063177      7A18JC8HW7A99/2[R65H/6Q*6Q*6I\
                           16  000011400600 067065067060 072122105115 072116117127 040107105124      009^606YV6Y+7AB8R:7A>9ZG4178RD
                           21  040105106106 105103124111 126105040101 104104122105 123123177177      4158/68R3[J9[/54118J4[A5[IC\Z\
                           26  000013000600 067065070060 072114104101 072117105101 054066072107      00#0606YV70+7A@8J17A\8R15^W7A7
                           31  105124040117 120105122101 116104040105 106106105103 124111126105      8RD41\[15[A19/44158/68R3[J9[/5
                           36  040101104104 122105123123 000010400600 067065071060 072123124103      4118J4[A5[IC008^606YV78+7AC[J3
                           41  101072103102 054060064072 123101126105 040103110101 122040120101      88 8I25^+6^ [I1[/5413911[ ^[11
                           46  122124177177 000007600600 067066060060 072124123130 061072117122      [AD\Z\007+606YW60+7AD[IH68 9ZB
                           51  107122105114 072122105114 101124111126 111132105177 000004400600      8ZB8R@7AB8R@89D99F99&8R\004^60
                           56  067066061060 072101122107 072116117122 107067177177 000010400600      6YW68+7A1[A77A>9ZB8YX\Z\008^60
                           61  067066062060 072123124103 121072103102 054067060072                   6YW6 +7AC[J3[8 8I25^X60
```

Figure A-4. Random Dump

```
$       FUTIL    01,,REW/01/,DDUMP/20R/
                          TAPE LABEL FILECODE - 01
                      1   *GE  600 BTL GE PHX 24609 24609  0001 73102    000                              *
 BLK#  REC#      RCW(L) WRD#                                     LOGICAL DUMP FILE#    1 FILECODE 01
   1     1       000321   1    *F1 REPORT A 000001\30000*
         2       000322   1    *F1 REPORT B 000001\20000*
         3       000323   1    *F1 REPORT C 000001\10000*
         4       000321   1    *F1 REPORT A 000002\30000*
   2     5       000322   1    *F1 REPORT B 000002\20000*
         6       000323   1    *F1 REPORT C 000002\10000*
         7       000321   1    *F1 REPORT A 000003\30000*
         8       000322   1    *F1 REPORT B 000003\20000*
   3     9       000323   1    *F1 REPORT C 000003\10000*
        10       000321   1    *F1 REPORT A 000004\30000*
        11       000322   1    *F1 REPORT B 000004\20000*
        12       000323   1    *F1 REPORT C 000004\10000*
   4    13       000321   1    *F1 REPORT A 000005\30000*
        14       000322   1    *F1 REPORT B 000005\20000*
        15       000323   1    *F1 REPORT C 000005\10000*
        16       000321   1    *F1 REPORT A 000006\30000*
   5    17       000322   1    *F1 REPORT B 000006\20000*
        18       000323   1    *F1 REPORT C 000006\10000*
        19       000321   1    *F1 REPORT A 000007\30000*
        20       000322   1    *F1 REPORT B 000007\20000*
```

Figure A-5. Logical DDUMP

```
$       FFILE   01,PHYREC
$       FUTIL   01,,REW/01/,DDUMP/10R/,REW/01/
                                    TAPE LABEL FILECODE - 01                        DENS   HIGH   MODE BINARY

                                  1   *GE   600 BTL GE PHX 24609 24609   0001 73102   000                                          *

BLK#        MODE     CC    WRD#                                    PHYREC  DUMP FILE#      1 FILECODE 01                 DENS  HIGH

  1        BINARY             1   *00100D004C3AF1 REPORT A 000001\3000000403BF1  REPORT B 000001\2000000403CF1 REPORT C 000001\10000
                             17    00403AF1 REPORT A 000002\30000*

  2                           1   *00200D00403BF1 REPORT B 000002\2000000403CF1  REPORT C 000002\1000000403AF1 REPORT A 000003\30000
                             17    00403BF1 REPORT B 000003\20000*

  3                           1   *00300D00403CF1 REPORT C 000003\1000000403AF1  REPORT A 000004\3000000403BF1 REPORT B 000004\20000
                             17    00403CF1 REPORT C 000004\10000*

  4                           1   *00400D00403AF1 REPORT A 000005\3000000403BF1  REPORT B 000005\2000000403CF1 REPORT C 000005\10000
                             17    00403AF1 REPORT A 000006\30000*

  5                           1   *00500D00403BF1 REPORT B 000006\2000000403CF1  REPORT C 000006\1000000403AF1 REPORT A 000007\30000
                             17    C0403BF1 REPORT B 000007\20000*

  6                           1   *00600D00403CF1 REPORT C 000007\1000000403AF1  REPORT A 000008\3000000403BF1 REPORT B 000008\20000
                             17    00403CF1 REPORT C 000008\10000*

  7                           1   *00700D00403AF1 REPORT A 000009\3000000403BF1  REPORT B 000009\2000000403CF1 REPORT C 000009\10000
                             17    00403AF1 REPORT A 000010\ 0000*

  8                           1   *00800D00403BF1 REPORT B 000010\2000000403CF1  REPORT C 000010\1000000403AF1 REPORT A 000011\30000
                             17    00403BF1 REPORT B 000011\20000*

  9                           1   *00900D00403CF1 REPORT C 000011\1000000403AF1  REPORT A 000012\3000000403BF1 REPORT B 000012\20000
                             17    00403CF1 REPORT C 000012\10000*

 10                           1   *00[00D00403AF1 REPORT A 000013\3000000403BF1  REPORT B 000013\2000000403CF1 REPORT C 000013\10000
                             17    00403AF1 REPORT A 000014\30000*
```

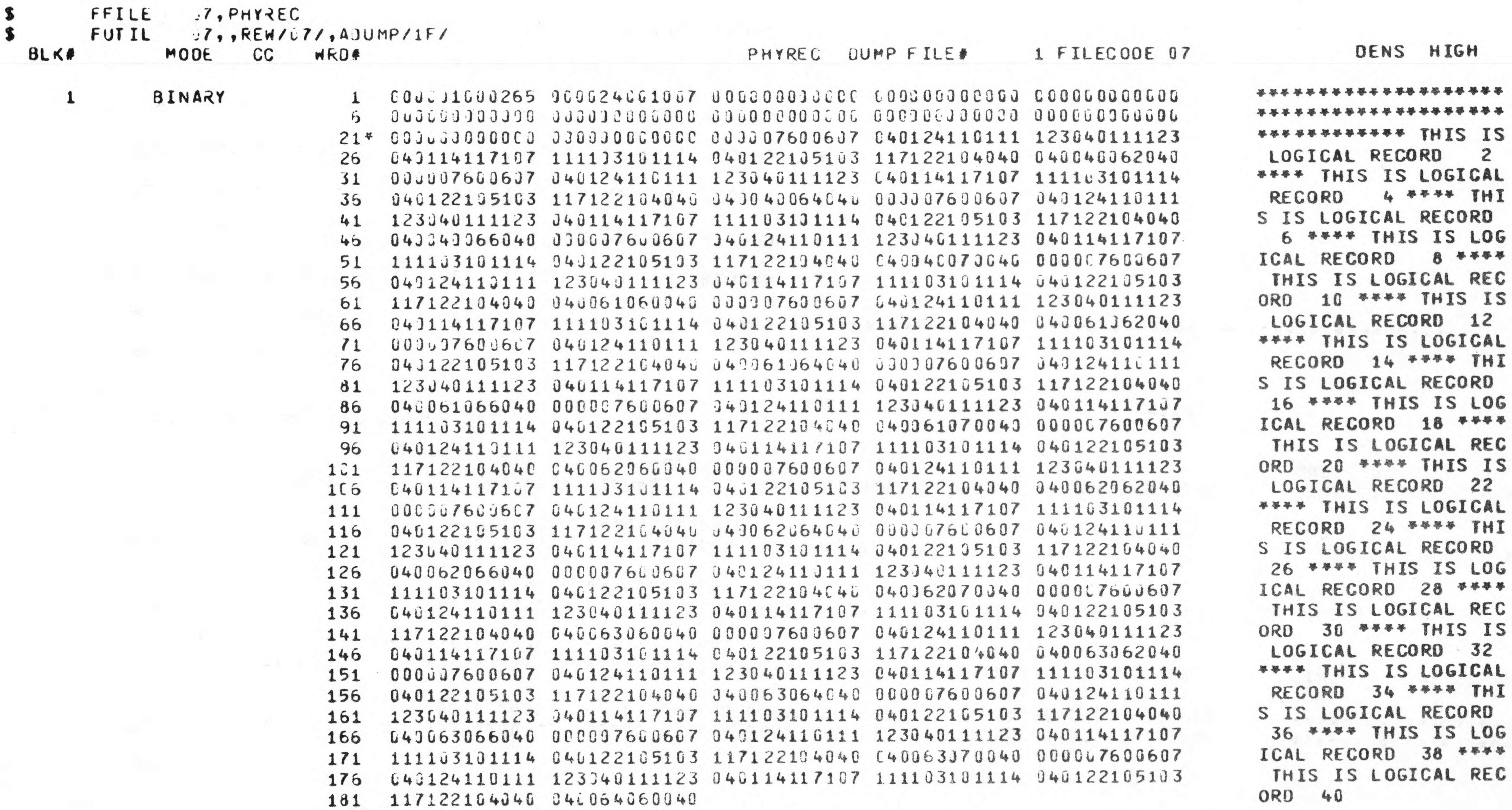
Figure A-6. PHYREC DDUMP

```
$     FFILE   F1,RANDOM/0
$     FUTIL   F1,,REW/F1/,DDUMP/6R/,REW/F1/
       REC#              WRD#                               RANDOM  DUMP FILE#     1 FILECODE F1

        1                   1   *00104\00D05000100000000[000000000000000000000000000000000000000000000000000000000000000000000000
                           17    0000000000000000000000000000000000004Q0504++60/6119J@41D911[^^8Z@99D[J5[AC419[H^9/\[^^8Z@99D[J5
                           33    [ ;1QR60+6 +88^[9E9939H^8/\#0^9AE9R [H^9ZF8RB41D91541@89&#8^8J\8Y;1R860+6H+9/\[Y^99C41D91541D99:
                           49    8Q^8/\[ ^89@9^^8Z\9Z441:8R>41D9Y^8I\9R541D9Y^[J88Q^8998^^9Z641D91599B41 89B[JI5+:3H+60U61 89E9^^*

        2                   1   *99C4199+^9J99I54169JI9/>5+:9Q+60V61\9/541\8+^[J88Q^9R199>41389E[I5[H^9Z641D8R59+)8978Q^8938I98J5
                           17    9/D[H^99C418[R78Z99/741D91541G[A\9/7413[R8[/55+:58+60W61D919[H^99C41D9154159/441\8+^[J88Q^8/E[J9
                           33    9J9[JI41D8RC[^:00<000000000000000000000000000000000000000000000000000000000000000000000000000000
                           49    000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000*

        3                   1   *000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                           49*   000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000*

        4                   1   *000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                           49*   000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000*

        5                   1   *000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
                           49*   000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000\00*

        6                   1   *00D04=006^606YV6Q+7A18JC8HZ7A:9939/E9Q*68*68+\Z\006+606YV6++7A18JC8HW7A99/2[R65H/6Q*6Q*6I\009^60
                           17    6YV6Y+7AB8R:7A>9ZG4178RD4158/68R3[J9[/54118J4[A5[IC\Z\00#0606YV70+7A@8J17A\8R15^W7A78RD41\[15[A1
                           33    9/44158/68R3[J9[/54118J4[A5[IC008^606YV78+7AC[J388_8I25^+6^_[I1[/5413911[ ^[11[AD\Z\007+606YW60+
                           49    7AD[IH68_9ZB8ZB8R@7AB8R@89D99F99&8R\004^606YW68+7A1[A77A>9ZB8YX\Z\008^606YW6 +7AC[J3[8 8I25^X68 *
```

Figure A-7. Random DDUMP

```
$      FUTIL    07,,RWD/07/,ADUMP/5R/
  REC#                     WRD#                                  ASCII   DUMP FILE#     1 FILECODE 07

       BCW   000001000265

     1 RCW   000024001007     1  000000000000 000000000000 000000000000 000000000000 000000000000        ********************
                             16* 000000000000 000000000000 000000000000 000000000000 000000000000        ********************

     2 RCW   000007600607     1  040124110111 123040111123 040114117107 111103101114 040122105103         THIS IS LOGICAL REC
                              6  117122104040 040040062040                                               ORD   2

     3 RCW   000007600607     1  040124110111 123040111123 040114117107 111103101114 040122105103         THIS IS LOGICAL REC
                              6  117122104040 040040064040                                               ORD   4

     4 RCW   000007600607     1  040124110111 123040111123 040114117107 111103101114 040122105103         THIS IS LOGICAL REC
                              6  117122104040 040040066040                                               ORD   6

     5 RCW   000007600607     1  040124110111 123040111123 040114117107 111103101114 040122105103         THIS IS LOGICAL REC
                              6  117122104040 040040070040                                               ORD   8
$      FUTIL    07,,RWD/07/,AADUMP/10R/
  REC#                                                           ASCII   DUMP FILE#     1 FILECODE 07

       BCW   000001000265

     1 RCW   000024001007         00 ********************************************************************************

     2 RCW   000007600607         27  THIS IS LOGICAL RECORD   2_

     3 RCW   000007600607         27  THIS IS LOGICAL RECORD   4_

     4 RCW   000007600607         27  THIS IS LOGICAL RECORD   6_

     5 RCW   000007600607         27  THIS IS LOGICAL RECORD   8_

     6 RCW   000007600607         27  THIS IS LOGICAL RECORD  10_

     7 RCW   000007600607         27  THIS IS LOGICAL RECORD  12_

     8 RCW   000007600607         27  THIS IS LOGICAL RECORD  14_

     9 RCW   000007600607         27  THIS IS LOGICAL RECORD  16_

    10 RCW   000007600607         27  THIS IS LOGICAL RECORD  18_
```

Figure A-8. Logical ADUMP and AADUMP



```
$       FFILE   07,PHYREC
$       FUTIL   07,,REW/07/,ADUMP/1F/
  BLK#        MODE    CC   WRD#                                           PHYREC   DUMP FILE#      1 FILECODE 07              DENS  HIGH

    1         BINARY          1  000001000265 000024001007 000000000000 000000000000 000000000000    ********************
                              6  000000000000 000000000000 000000000000 000000000000 000000000000    ********************
                             21* 000000000000 000000000000 000007600607 040124110111 123040111123    ************ THIS IS
                             26  040114117107 111103101114 040122105103 117122104040 040040062040     LOGICAL RECORD   2
                             31  000007600607 040124110111 123040111123 040114117107 111103101114    **** THIS IS LOGICAL
                             36  040122105103 117122104040 040040064040 000007600607 040124110111     RECORD   4 **** THI
                             41  123040111123 040114117107 111103101114 040122105103 117122104040    S IS LOGICAL RECORD
                             46  040040066040 000007600607 040124110111 123040111123 040114117107      6 **** THIS IS LOG
                             51  111103101114 040122105103 117122104040 040040070040 000007600607    ICAL RECORD   8 ****
                             56  040124110111 123040111123 040114117107 111103101114 040122105103     THIS IS LOGICAL REC
                             61  117122104040 040061060040 000007600607 040124110111 123040111123    ORD  10 **** THIS IS
                             66  040114117107 111103101114 040122105103 117122104040 040061062040     LOGICAL RECORD  12
                             71  000007600607 040124110111 123040111123 040114117107 111103101114    **** THIS IS LOGICAL
                             76  040122105103 117122104040 040061064040 000007600607 040124110111     RECORD  14 **** THI
                             81  123040111123 040114117107 111103101114 040122105103 117122104040    S IS LOGICAL RECORD
                             86  040061066040 000007600607 040124110111 123040111123 040114117107      16 **** THIS IS LOG
                             91  111103101114 040122105103 117122104040 040061070040 000007600607    ICAL RECORD  18 ****
                             96  040124110111 123040111123 040114117107 111103101114 040122105103     THIS IS LOGICAL REC
                            101  117122104040 040062060040 000007600607 040124110111 123040111123    ORD  20 **** THIS IS
                            106  040114117107 111103101114 040122105103 117122104040 040062062040     LOGICAL RECORD  22
                            111  000007600607 040124110111 123040111123 040114117107 111103101114    **** THIS IS LOGICAL
                            116  040122105103 117122104040 040062064040 000007600607 040124110111     RECORD  24 **** THI
                            121  123040111123 040114117107 111103101114 040122105103 117122104040    S IS LOGICAL RECORD
                            126  040062066040 000007600607 040124110111 123040111123 040114117107      26 **** THIS IS LOG
                            131  111103101114 040122105103 117122104040 040062070040 000007600607    ICAL RECORD  28 ****
                            136  040124110111 123040111123 040114117107 111103101114 040122105103     THIS IS LOGICAL REC
                            141  117122104040 040063060040 000007600607 040124110111 123040111123    ORD  30 **** THIS IS
                            146  040114117107 111103101114 040122105103 117122104040 040063062040     LOGICAL RECORD  32
                            151  000007600607 040124110111 123040111123 040114117107 111103101114    **** THIS IS LOGICAL
                            156  040122105103 117122104040 040063064040 000007600607 040124110111     RECORD  34 **** THI
                            161  123040111123 040114117107 111103101114 040122105103 117122104040    S IS LOGICAL RECORD
                            166  040063066040 000007600607 040124110111 123040111123 040114117107      36 **** THIS IS LOG
                            171  111103101114 040122105103 117122104040 040063070040 000007600607    ICAL RECORD  38 ****
                            176  040124110111 123040111123 040114117107 111103101114 040122105103     THIS IS LOGICAL REC
                            181  117122104040 040064060040                                           ORD  40
```

Figure A-9. PHYREC ADUMP

```
$      FFILE   07,PHYREC
$      FUTIL   07,,REW/07/,AADUMP/1R/
  BLK#       MODE   CC                                      PHYREC  DUMP FILE#     1 FILECODE 07            DENS  HIGH

     1      BINARY               ******************************************************************************************** THI
                                 S IS LOGICAL RECORD   2 **** THIS IS LOGICAL RECORD   4 **** THIS IS LOGICAL RECORD   6 **** THI
                                 S IS LOGICAL RECORD   8 **** THIS IS LOGICAL RECORD  10 **** THIS IS LOGICAL RECORD  12 **** THI
                                 S IS LOGICAL RECORD  14 **** THIS IS LOGICAL RECORD  16 **** THIS IS LOGICAL RECORD  18 **** THI
                                 S IS LOGICAL RECORD  20 **** THIS IS LOGICAL RECORD  22 **** THIS IS LOGICAL RECORD  24 **** THI
                                 S IS LOGICAL RECORD  26 **** THIS IS LOGICAL RECORD  28 **** THIS IS LOGICAL RECORD  30 **** THI
                                 S IS LOGICAL RECORD  32 **** THIS IS LOGICAL RECORD  34 **** THIS IS LOGICAL RECORD  36 **** THI
                                 S IS LOGICAL RECORD  38 **** THIS IS LOGICAL RECORD  40
```

Figure A-10. PHYREC AADUMP

INDEX

CUT ALONG LINE

HONEYWELL INFORMATION SYSTEMS
Technical Publications Remarks Form

TITLE: SERIES 60(LEVEL 66)/6000 UTILITY

ORDER NO. DD12, REV. 0

DATED APRIL 1974

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

Your comments will be promptly investigated by appropriate technical personnel and action will be taken as required. If you require a written reply, check here and furnish complete mailing address below. □

FROM: NAME ____________________ DATE ____________

TITLE ____________________

COMPANY ____________________

ADDRESS ____________________

____

CUT ALONG LINE

PLEASE FOLD AND TAPE –
NOTE: U. S. Postal Service will not deliver stapled forms

FOLD ALONG LINE

FIRST CLASS
PERMIT NO. 39531
WALTHAM, MA
02154

Business Reply Mail
Postage Stamp Not Necessary if Mailed in the United States

Postage Will Be Paid By:

HONEYWELL INFORMATION SYSTEMS
200 SMITH STREET
WALTHAM, MA 02154

ATTENTION: PUBLICATIONS, MS 486

FOLD ALONG LINE

Honeywell

Honeywell

Honeywell Information Systems
In the U.S.A.: 200 Smith Street, MS 486, Waltham, Massachusetts 02154
In Canada: 2025 Sheppard Avenue East, Willowdale, Ontario M2J 1W5
In Mexico: Avenida Nuevo Leon 250, Mexico 11, D.F.

20834, 3578, Printed in U.S.A.