File Number S360-32
Order No. GC24-3465-8

DOS
TOS

IBM

Systems Reference Library

# DOS and TOS Utility Programs

This reference publication describes to programmers the use of the IBM Disk and Tape Operating Systems Utility Programs. Both file-to-file and special-purpose utilities are included. The programs described are:

360N-UT-461

Assign Alternate Track-Disk
Card to Printer and/or Punch
Card to Disk
Clear Disk
Copy-Disk to Disk
Copy and Restore-Disk to Card
Disk to Card
Disk to Disk
Disk to Printer
Initialize Disk
VTOC Display

360N-UT-462

Card to Tape
Copy and Restore-Disk or Data Cell to Tape
Data Cell to Tape
Disk to Tape
Initialize Tape
Tape to Card
Tape Compare
Tape to Data Cell
Tape to Disk
Tape to Printer
Tape to Tape

360N-UT-463

Assign Alternate Track-Data Cell
Clear Data Cell
Data Cell to Data Cell
Data Cell to Disk
Data Cell to Printer
Disk to Data Cell
Initialize Data Cell

The reader should be familiar with these SRL publications for the IBM System/360 Disk and Tape Operating Systems: IBM System/360 TOS, System Control and System Service Programs, GC24-5034; DOS System Control and Service, GC24-5036. For titles and abstracts of other associated publications, see the IBM System/360 and System/370 Bibliography, GA22-6822.

**DOS Release 26.1**

Ninth Edition (August, 1973)

This is a reprint of GC24-3465-7 incorporating changes released in Technical Newsletter GN33-8738 (dated January 1, 1973).

This edition applies to Release 26.1 of the IBM Disk Operating System and to all subsequent releases until otherwise indicated in new edition or technical newsletters. Before using this publication in connection with the operation of IBM systems, consult the latest System/360 and System/370 SRL Newsletter, GN20-0360, for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

Comments concerning the contents of this publication may be addressed to IBM Laboratory, Publications Dept., P.O. Box 24, Uithoorn, The Netherlands. Comments become the property of IBM.

© Copyright International Business Machines Corporation 1965, 1968, 1969, 1970, 1971

## Control Statement Information

Some basic terminology is used when illustrating control statements.

1. A parameter is a variable with its identifying character that is given a constant value for a specific purpose (i.e....,Sx,... where Sx is a parameter possibly within a string of parameters, S being the identifier and x the variable).

2. DASD means a Direct Access Stcrage Device such as a disk or data cell drive.

3. Uppercase letters and punctuation marks represent information that must be coded exactly as shown.

4. Lowercase letters represent information that must be supplied by the programmer. The letter b always indicates one blank space. Fcr parameter variables, lowercase letters represent constants that must be supplied and the lowercase x represents an optional parameter.

5. Options contained within braces { } represent alternatives, one of which must be chosen.

6. An ellipsis (a series of three periods) indicates that a variable number of items may be included.

7. All references to disk refer to the IBM 2311 and IBM 2314 and 2319 unless otherwise stated.

## Supplementary Reading References

The reader is directed to the following SRL publications for supplementary reference:

DISK OPERATING SYSTEM

DOS Supervisor and I/O Macros, GC24-5037.

DOS Data Management Concepts, GC24-3427.

DOS System Generation, GC24-5033.

DOS Messages, GC24-5074.

TAPE OPERATING SYSTEM

IBM System/360 Tape Operating System, Supervisor and Input/Output Macros, GC24-5035.

IBM System/360 Tape Operating System, Data Management Concepts, GC24-3430.

IBM System/360 Tape Operating System, System Generation and Maintenance, GC24-5012.

# Contents


**Introduction** . . . . . 7
Description . . . . . 7
File-to-File Utility Programs . . . . 7
Special-Purpose Utility Programs . . . 7
Common Characteristics . . . . . 7
Machine Requirements . . . . . 8
Control Statements . . . . . 8
Job Control . . . . . 8
Linkage Editor . . . . . 9
Checkpoint Records . . . . . 13

**Explanations of the Utility Specifications** . . . 14
File-to-File Utilities . . . . . 14
Label Checking . . . . . 14
Nonstandard and User Label Handling . 14
Utility Message Routines . . . . . 15
Multifile Volume (Tape) . . . . . 16
Multivolume Files (Tape) . . . . . 16
Record Skipping . . . . . 16
Sequence Numbering . . . . . 16
Binary Records . . . . . 17
Printer Output . . . . . 17
Data Display . . . . . 17
Data List . . . . . 17
Available I/O Area . . . . . 17
Minimum I/O Area . . . . . 17
I/O Area Assignment . . . . . 18
First-Character Forms Control . . . . . 18
Type A . . . . . 19
Type B . . . . . 20
Type C . . . . . 20
Type D . . . . . 20
SYSLST/SYS005 Carriage Control . . . . 21
First-Character Stacker-Select Control . 21
Utility Modifier Statement . . . . . 21
Parameter Combinations . . . . . 25
Field Select Statement . . . . . 25
Pack . . . . . 26
Unpack . . . . . 26
Hexadecimal . . . . . 27
Copy Variable . . . . . 27
Heading Statement . . . . . 27
END Statement . . . . . 27
Examples . . . . . 27
Card to Tape . . . . . 27
Tape to Tape . . . . . 28
Disk to Disk . . . . . 28
Key Fields . . . . . 28
DASD Files With Key Fields (Fixed-Length Unblocked) . . . . . 28
DASD Files With Key Fields (Undefined) . . . . . 29
Recording Job Control Statements on Output File . . . . . 30

**Part 1. File—to—File Utility Programs** . 31

**Card to Disk** . . . . . 32
Utility Modifier Statement . . . . . 32
Field Select Statement . . . . . 32
Pack . . . . . 32
Unpack . . . . . 34
Control Statement Stream . . . . . 34

**Card to Printer and/or Punch** . . . . . 35
Card to Printer . . . . . 35
Display . . . . . 35
List . . . . . 35
Card to Punch . . . . . 35
Card to Printer and Punch . . . . . 35
Utility Modifier Statement . . . . . 35
Field Select Statement . . . . . 35
Pack . . . . . 36
Unpack . . . . . 36
Hexadecimal . . . . . 36
Control Statement Stream . . . . . 36

**Card to Tape** . . . . . 39
Utility Modifier Statement . . . . . 39
Field Select Statement . . . . . 39
Pack . . . . . 39
Unpack . . . . . 39
Control Statement Stream . . . . . 41

**Data Cell to Data Cell** . . . . . 42
Utility Modifier Statement . . . . . 42
Field Select Statement . . . . . 42
Pack . . . . . 44
Unpack . . . . . 44
Copy Variable . . . . . 44
Control Statement Stream . . . . . 44

**Data Cell to Disk** . . . . . 45
Utility Modifier Statement . . . . . 45
Field Select Statement . . . . . 45
Pack . . . . . 47
Unpack . . . . . 47
Copy Variable . . . . . 47
Control Statement Stream . . . . . 47

**Data Cell to Printer** . . . . . 48
Utility Modifier Statement . . . . . 48
Field Select Statement . . . . . 48
Unpack . . . . . 48
Hexadecimal . . . . . 48
Copy Variable . . . . . 50
Control Statement Stream . . . . . 50

**Data Cell to Tape** . . . . . 51
Utility Modifier Statement . . . . . 51
Field Select Statement . . . . . 51
Pack . . . . . 51
Unpack . . . . . 51
Copy Variable . . . . . 53
Control Statement Stream . . . . . 53

**Disk to Card** . . . . . 54
Sequence Numbering . . . . . 54
Utility Modifier Statement . . . . . 54
Field Select Statement . . . . . 54
Pack . . . . . 56
Unpack . . . . . 56
Control Statement Stream . . . . . 56

**Disk to Data Cell** . . . . . 57
Utility Modifier Statement . . . . . 57
Field Select Statement . . . . . 57
Pack . . . . . 58
Unpack . . . . . 58
Copy Variable . . . . . 58
Control Statement Stream . . . . . 58

**Disk to Disk** . . . . . 60
Utility Modifier Statement . . . . . 60
Field Select Statement . . . . . 60
Pack . . . . . 62
Unpack . . . . . 62
Copy Variable . . . . . 62
Control Statement Stream . . . . . 62

**Disk to Printer** . . . . . 63
Utility Modifier Statement . . . . . 63
Field Select Statement . . . . . 63
Unpack . . . . . 63
Hexadecimal . . . . . 63
Copy Variable . . . . . 65
Indexed Sequential Handling . . . . . 65
Control Statement Stream . . . . . 65

**Disk to Tape** . . . . . 66
Utility Modifier Statement . . . . . 66
Field Select Statement . . . . . 66
Pack . . . . . 66
Unpack . . . . . 68
Copy Variable . . . . . 68
Control Statement Stream . . . . . 68

**Tape to Card** . . . . . 69
Sequence Numbering . . . . . 69
Utility Modifier Statement . . . . . 69
Field Select Statement . . . . . 69
Pack . . . . . 69
Unpack . . . . . 71
Control Statement Stream . . . . . 71

**Tape to Data Cell** . . . . . 72
Utility Modifier Statement . . . . . 72
Field Select Statement . . . . . 72
Pack . . . . . 72
Unpack . . . . . 74
Copy variable . . . . . 74
Control Statement Stream . . . . . 74

**Tape to Disk** . . . . . 75
Utility Modifier Statement . . . . . 75
Field Select Statement . . . . . 75
Pack . . . . . 75
Unpack . . . . . 77
Copy Variable . . . . . 77
Control Statement Stream . . . . . 77

**Tape to Printer** . . . . . 78
Utility Modifier Statement . . . . . 78
Field Select Statement . . . . . 78
Unpack . . . . . 78
Hexadecimal . . . . . 80
Copy Variable . . . . . 80
Control Statement Stream . . . . . 80

**Tape to Tape** . . . . . 81
Utility Modifier Statement . . . . . 81
Field Select Statement . . . . . 81
Pack . . . . . 81
Unpack . . . . . 81
Copy Variable . . . . . 83
Control Statement Stream . . . . . 83

**Part 2. Special—Purpose Utility Programs** . . . 85

**Assign Alternate Track—Data Cell** . . . . 86
Assigning an Alternate Track . . . . . 86
Utility Modifier Statement . . . . . 87
Track Statement . . . . . 88
Control Statement Stream . . . . . 88

**Assign Alternate Track—Disk** . . . . . 89
Assigning an Alternate Track . . . . . 89
Changing Track-Condition Indication . 90
Utility Modifier Statement . . . . . 90
Track Statement . . . . . 92
Control Statement Stream . . . . . 92

**Clear Data Cell** . . . . . 93
Utility Modifier Statement . . . . . 93
END Statement . . . . . 94
Control Statement Stream . . . . . 94

**Clear Disk** . . . . . 95
Utility Modifier Statement . . . . . 95
END Statement . . . . . 96
Control Statement Stream . . . . . 96

**Copy and Restore Disk or Data Cell** . . . 97
Checkpoint and Restart Facility . . . 97
Copy Disk or Data Cell Programs . . . . 97
I/O Area . . . . . 98
Opening a Disk Pack or Data Cell . . . 98
Processing User Standard Labels on DASD Files . . . . . 98
Opening the Tape Volume . . . . . 98
Utility Modifier Statement . . . . . 98
Input Processing . . . . . 99
Restore Disk or Data Cell Programs . . .100
I/O Area . . . . . .100
Opening a Disk Pack or Data Cell for Restoring a File . . . . . .100
Opening a Disk Pack or Data Cell for Restoring a Volume . . . . . .100
Opening the Tape Volume(s) . . . . . .101
Closing the Tape Volume(s) . . . . . .101
Control Statement Stream . . . . . .101

**Initialize Data Cell** . . . . . .103
VTOC Label Checking . . . . . .103
Home Address Generation . . . . . .103
Track Descriptor (R0) Generation . . . .103
Surface Analysis and Initialization Verification . . . . . .103
Volume Label Creation . . . . . .103
IPL and VTOC Format Creation . . . . . .103
IPL Format Creation . . . . . .103
VTOC Format Creation . . . . . .103
Utility Modifier Statement . . . . . .104
Label Control Set . . . . . .104
VTOC Control Card . . . . . .104
Volume Label Control Card . . . . . .104
END Card . . . . . .105
Control Statement Stream . . . . . .105

**Initialize Disk** . . . . . .106
VTOC Label Checking . . . . . .106
Home Address Generation . . . . . .106
Surface Analysis and Track Descriptor Record (R0) Generation . . . . . .106
Volume Label Creation . . . . . .107
IPL and VTOC Creation . . . . . .107
IPL Format Creation . . . . . .107
VTOC Format Creation . . . . . .107
Utility Modifier Statement . . . . . .107
Label Control Set . . . . . .107
VTOC Control Card . . . . . .109
Volume Label Control Card . . . . . .109
END Card . . . . . .110
Control Statement Stream . . . . . .110

**Initialize Tape** . . . . . .111
Job Control Cards . . . . . .111
Utility Control Statements . . . . . .111
Control Statement Stream . . . . . .113

**Tape Compare** . . . . . .114
Restart Procedures . . . . . .114
Label Processing . . . . . .114
Nonstandard or Unlabeled Files . . . . .115
Job Control Statements . . . . . .115
Linkage Editor . . . . . .115
Utility Control Statement . . . . . .116
User Exit Routine . . . . . .116
Control Statement Stream . . . . . .117

**VTOC Display** . . . . . .118
Linkage Edit Procedure . . . . . .118
Control Cards . . . . . .118

**Appendix A. Module Contents** . . . . .120

**Appendix B. Printer Output** . . . . . .125

**Appendix C. File—to—File Program Messages** . .127

**Appendix D. Tape Compare, Copy/Restore, and Initialize Tape Program Messages** . . . . . .137

**Appendix E. Clear Data Cell and Clear Disk Program Messages** . . . . . .139

**Appendix F. Operator Communication Messages** . . . . . .141
File-to-File Messages . . . . . .141
Tape Compare Messages . . . . . .141
Copy/Restore Disk or Data Cell Messages 148
Initialize Disk and Data Cell Utility Messages . . . . . .155
Alternate Track Assign Utility Messages 159
Initialize Tape Messages . . . . . .169
VTOC Display Messages . . . . . .169

**Index** . . . . . .172

# Figures

Figure 1. Job Control Statements Used in Each Program . . . . . 9
Figure 2. Job Control Statement File Names and Assignments . . . . . 10
Figure 3. Phase and Module Names for the File-to-File Programs . . . . . 11
Figure 4. Phase and Module Names for the Special Purpose Programs . . . . . 12
Figure 5. Answer Byte or Entry Point Completions . . . . . 16
Figure 6. Utility Modifier Statement Parameters (Part 1 of 2) . . . . . 22
Figure 7. Card-to-Disk Utility Modifier Statement . . . . . 33
Figure 8. Card-to-Printer and/or Punch Utility Modifier Statement (Part 1 of 2) . . . . . 37
Figure 9. Card-to-Tape Utility Modifier Statement . . . . . 40
Figure 10. Data Cell-to-Data Cell Utility Modifier Statement . . . . . 43
Figure 11. Data Cell-to-Disk Utility Modifier Statement . . . . . 46
Figure 12. Data Cell-to-Printer Utility Modifier Statement (Part 1 of 2) . . . . . 49
Figure 13. Data Cell-to-Tape Utility Modifier Statement . . . . . 52
Figure 14. Disk-to-Card Utility Modifier Statement . . . . . 55
Figure 15. Disk-to-Data Cell Utility Modifier Statement . . . . . 59
Figure 16. Disk-to-Disk Utility Modifier Statement . . . . . 61
Figure 17. Disk-to-Printer Utility Modifier Statement (Part 1 of 2) . . . 64
Figure 18. Disk-to-Tape Utility Modifier Statement . . . . . 67
Figure 19. Tape-to-Card Utility Modifier Statement . . . . . 70
Figure 20. Tape-to-Data Cell Utility Modifier Statement (Part 1 of 2) . . . 73
Figure 21. Tape-to-Disk Utility Modifier Statement (Part 1 of 2) . . . 76
Figure 22. Tape-to-Printer Utility Modifier Statement (Part 1 of 2) . . . 79
Figure 23. Tape-to-Tape Utility Modifier Statement . . . . . 82
Figure 24. Assign Alternate Track-Disk Utility Modifier Statement . 91
Figure 25. Initialize Disk Utility Modifier Statement . . . . . .108

# Introduction

Whatever the specific uses of a data processing system, certain unique operations exist that must be performed frequently. These operations may differ in detail, depending on the particular machine configuration and data format for the individual user, but the essential function remains the same. Generalized routines designed to satisfy specific functions would ease the burden of programming these operations. These routines must be flexible enough, however, to allow the user to assign the specifications of his particular problem.

One type of program that meets these requirements are utility programs. They are designed to assist the user in day-to-day operation of his installation. With these programs the user can perform certain frequently required operations, such as transferring disk-storage files to cards or tape and printing out areas of tape or disk for program-testing purposes, without programming effort.

## Description

Two groups of programs are described in this publication. The first group, the file-to-file utility programs, transfers the contents of a single file from an input medium to an output medium. The second group, the special-purpose utility programs, performs a particular function not likely to be used every day.

FILE-TO-FILE UTILITY PROGRAMS

Seventeen file-to-file programs can transfer files from input mediums to output mediums. These programs are:

Card to Disk
Card to Printer and/or Punch
Card to Tape
Data Cell to Data Cell
Data Cell to Disk
Data Cell to Printer
Data Cell to Tape
Disk to Card
Disk to Data Cell
Disk to Disk
Disk to Printer
Disk to Tape
Tape to Card
Tape to Data Cell
Tape to Disk
Tape to Printer
Tape to Tape

SPECIAL-PURPOSE UTILITY PROGRAMS

Assign Alternate Track-Data Cell assigns an alternate track when a track has proven defective.

Assign Alternate Track-Disk assigns an alternate track when a track has proven defective.

Clear Data Cell clears one or more areas of a data cell and establishes preformatted tracks.

Clear Disk clears one or more areas of a disk and establishes preformatted tracks.

Copy and Restore-Disk to Card, Copy and Restore-Disk or Data Cell to Tape, and Copy-Disk to Disk copy an entire file or volume onto the specified media and restore the file or volume to its original media.

Initialize Data Cell initializes data cells and does surface analysis on them.

Initialize Disk initializes a disk pack and does surface analysis on it.

Initialize Tape initializes tapes with IBM or ANSI standard volume labels.

Tape Compare compares two files from two or more tapes to ensure that the files are identical.

VTOC Display displays the volume table of contents of one disk pack.

All thirty programs described are disk resident. Six of these disk resident utilities, however, are also available as tape resident utilities. These utilities are:

Card to Printer and/or Punch
Card to Tape
Tape to Card
Tape to Printer
Tape to Tape
Tape Compare

If the function of a disk resident and a tape resident utility function is identical, only one description is given.

COMMON CHARACTERISTICS

Each utility program handles a particular type of job. A symbolic assembly is not necessary for the operation of the program.

The utility programs process either consecutive or split-cylinder type files. For more information on these file arrangements, see the DOS Supervisor and I/O Macros publication listed in the preface. Output records from sequential files (consecutive or split-cylinder) produced by the System/360 Operating System can be processed by these utility programs, but these records must not be printed and punched.

Indexed sequential files are not generally supported by these utility programs. The only exceptions are noted in the disk-to-printer and the copy/restore programs.

All utility control cards shown in this publication begin with // in the first two columns, however, ./ is also acceptable.

To handle a specific job, the generalized program is modified by utility control cards. Control cards for all programs (except initialize tape) are free-form in that optional parameters can be punched in any order. The programs assume default values for most options when a choice is not indicated in a utility control card.

Control information is specified in a similar manner for all programs. When the same device is used with different programs, the control information related to the device is similar for all programs. Label handling and the control information related to input and output device assignment and description are compatible with IBM Disk and Tape Operating Systems IOCS.

## Machine Requirements

The minimum machine configuration required for these programs is an IBM System/360 or System/370 central processing unit with 16K bytes of main storage, however, the tape compare program requires 24K bytes of main storage. For control statement loading use one of these machines:

IBM 2520 Card Read Punch
IBM 2540 Card Read Punch
IBM 1442 Card Reader
IBM 2501 Card Reader
IBM 2311 Disk Storage Drive
IBM 2314 Direct Access Storage Facility
IBM 2319 Disk Storage
IBM 2400 Series Magnetic Tape Unit
IBM 3420 Magnetic Tape Unit*

For program residence use one of these machines:

IBM 2311 Disk Storage Drive
IBM 2314 Direct Access Storage Facility
IBM 2319 Disk Storage
IBM 2400 Series Magnetic Tape Unit

For program operation, input/output devices are required by the specific program. Supported devices include:

IBM 1442 Card Read Punch
IBM 2520 Card Read Punch
IBM 2540 Card Read Punch
IBM 2501 Card Reader
IBM 2321 Data Cell Drive
IBM 2311 Disk Storage Drive
IBM 2314 Direct Access Storage Facility
IBM 2319 Disk Storage
IBM 1403 Printer
IBM 1404 Printer (continuous-forms printing only)
IBM 1443 Printer
IBM 3211 Printer (no optional features)
IBM 2400 Series Magnetic Tape Unit (with or without the 7-track feature)
IBM 3420 Magnetic Tape Unit*

Note: For Copy and Restore Disk to Tape with the 7-track feature, the data conversion feature is required.

For logging and error messages use one of these machines:

IBM 2311 Disk Storage Drive
IBM 2314 Direct Access Storage Facility
IBM 2319 Disk Storage
IBM 1403 Printer
IBM 1404 Printer (continuous forms only)
IBM 1443 Printer
IBM 3211 Printer (no optional features)
IBM 1052 Printer-Keyboard
IBM 2400 Series Magnetic Tape Unit
IBM 3420 Magnetic Tape Unit*

## Control Statements

JOB CONTROL

Job control statements related to channel and unit assignment, label processing, and physical-device description are used with the utility programs. For information on job control statements see the System Control and Service publications as listed on the front cover of this publication. The required job control statements for running each of these programs are given in Figure 1; the entries for specific fields unique to the utility programs are shown in Figure 2. The new simplified job control label cards are used in job control input stream examples throughout this publication.

* Note: 3420 support is provided only to a level equal to that of the 2400 series. This means that only 6 bytes of sense information are available, instead of the full 20.

| | FILE TO FILE PROGRAMS | ASSIGN ALTERNATE TRACK-DISK OR DATA CELL, INITIALIZE DISK OR DATA CELL, INITIALIZE TAPE | CLEAR DISK, CLEAR DATA CELL | COPY AND RESTORE DISK OR DATA CELL PROGRAMS | TAPE COMPARE | VTOC DISPLAY |
|---|---|---|---|---|---|---|
| JOB | Required | Required | Required | Required | Required | Required |
| LBLTYP | Required for link edit if tape label checking. | Not Used | Not Used | Required for link edit if tape label checking or copying an NSD file. | Not Used | Required if output is labeled tape. |
| TLBL | Required if tape label processing. | Not Used | Not Used | Required if tape label processing. | Not Used | Required if output is labeled tape. |
| DLBL | Required for DASD files. | Not Used | Required for DASD files. | Required for copy file and restore functions. Not used for copy volume function. | Not Used | Required if output is disk. |
| EXTENT | Required for DASD. | Not Used | Required for DASD. | Required for copy file and restore functions. Not used for copy volume function. | Not Used | Required if output is disk. |
| ASSGN | Required if devices are different than those at IPL time. | Required if devices are different than those at IPL time. | Required if devices are different than those at IPL time. | Required if devices are different than those at IPL time. | Required if devices are different than those at IPL time. | Required if devices are different than those at IPL time. |
| UPSI | Optional | Required for System/360 Model 30 and Model 40 Emulators (program numbers EU484, EU485) and for System/370 Emulators of 1401, 1440, 1460, and 1410/7010 programs (program number EU490) when initializing disk. | Required for System/360 Model 30 and 40 Emulators (program numbers EU484, EU485) and for System/370 Emulators of 1401, 1440, 1460, and 1410/7010 programs (program number EU490) when clearing cylinder 200 of a disk. | Optional | Optional | Not Used |
| EXEC | Required | Required | Required | Required | Required | Required |
| /* | Required for card input programs.* | Used only if update records are present. | Not Used | Not Used | Not Used | Not Used |
| /& | Required | Required | Required | Required | Required | Required |

* This card must immediately follow the data cards for card input programs. In addition card columns 3-80 of the card must be entirely blank, otherwise the card will be ignored and treated as data.

Figure 1. Job Control Statements Used in Each Program

LINKAGE EDITOR

When a program has not been cataloged in the core image library, the link editing phase must be performed. For detailed explanations of the linkage editor, see the System Control and Service publications listed on the front cover of this publication. For applicable linkage editor job control statements, refer to the System Generation publications listed in the preface of this publication.

All programs are loaded into main storage by the system loader. The programs may be link-edited to be executed in the background area or in a batched-job foreground area. The one exception is the initialize data cell program which will not run in the foreground if the multiple cells option is to be used.

When file-to-file programs are cataloged in the core image or relocatable library, the phase or module names must be used.

| | FILE TO FILE PROGRAMS | ASSIGN ALTERNATE TRACK DISK OR DATA CELL, INITIALIZE DISK OR DATA CELL OR TAPE | CLEAR DISK, CLEAR DATA CELL | COPY AND RESTORE DISK OR DATA CELL PROGRAMS | TAPE COMPARE | VTOC DISPLAY |
|---|---|---|---|---|---|---|
| Filename (TLBL or DLBL) | UIN for input file. UOUT for output file. | Not Used | UOUT | UIN for input devices. UOUT for output devices. UCHKPT for labeled checkpoint file. | Not Used | UOUT |
| ASSGN Device for Logging Operator Messages | SYSLOG | SYSLOG | SYSLOG | SYSLOG | SYSLOG | SYSLOG |
| ASSGN Utility Control Statement Input Device | SYSIPT | SYSIPT | SYSIPT | SYSIPT | SYSIPT | Not Used |
| ASSGN Device for Logging Programmer Messages | SYSLST | SYSLST (not applicable to Initialize Tape) | SYSLST | SYSLST | SYSLST | Not Used |
| ASSGN Primary Tape and Card, Input and Alternate Tape Input | SYS004 | Not Used | Not Used | SYS004 | SYS004 | Not Used |
| ASSGN Primary Tape and Printer Output and Alternate Tape Output | SYS005 | For Initialize Tape only: SYS000 required SYS001-SYS015 optional | Not Used | SYS005 | SYS005 | SYS005 |
| ASSGN Linkage Editor * | SYSLNK SYS001 SYS002 ** | SYSLNK SYS001 | SYSLNK SYS001 | SYSLNK SYS001 | SYSLNK SYS001 SYS002 ** | SYSLNK SYS001 |
| ASSGN Private Core Image Library ***** | SYSCLB | SYSCLB | SYSCLB | SYSCLB | SYSCLB | Not Used |
| ASSGN Card Output Device | SYS006 | Not Used | Not Used | SYS006 | Not Used | Not Used |
| ASSGN DASD Input and/or Output Device *** | SYS000-SYSnnn | SYS000 | SYS000-SYSnnn | SYS000-SYSnnn **** | Not Used | SYS004 SYS005 |

* This programmer unit is available when not in use by the linkage editor.
** This unit is needed for TOS only.
*** SYSnnn can be no greater than the greatest physical unit block assigned and must not conflict with the assignment of any other device.
**** SYS004 is required for the copy volume function.
***** SYSCLB is required when link-editing to or executing from a private core image library.

Figure 2. Job Control Statement File Names and Assignments

Figure 3 gives the phase and module names required when link-editing the file-to-file utility programs. Each of the 17 programs is contained in five phases. Phases 2 (module name IJWGEN) and 5 (module name IJWLAB) are the same for all programs and need only be loaded in two relocatable modules. Figure 4 gives the phase and module names for the special-purpose utility programs. The PHASE card must be omitted when using a book name to link-edit the special-purpose utilities (exception, see tape compare program). See Appendix A for the contents of the modules for all utility programs.

The following are sample control statements that can be used to execute a program. A typical example of the control cards used when executing a tape-to-tape program that is resident in the relocatable library is:

```
//bJOB
     user-defined job name.

//bASSGN
     assigns the input and output devices.

//bOPTION LINK
     indicates that the program is to be
     link-edited.

bINCLUDE IJWTT
     identifies the tape-to-tape modules to
     be link-edited.

bPHASEbTPTP5,IJWTTCS2,NOAUTO
     gives the name of the last phase of
     the program and the main-storage
     address where it is to be loaded by
     using the operand in the previous
     control card followed by CS2.

bINCLUDE IJWLAB
     link-edits the dummy label module. If
     the operand is omitted from this
     statement, the text of the user's
     routine must be present on SYSIPT and
     followed by the /* statement. (If
     SYSRDR and SYSIPT are the same device,
     the user's routine must be inserted
     after this INCLUDE statement.) If a
     user's routine is supplied from the
     relocatable library, that module's
     unique name must be entered in place
     of the IJWLAB operand. (Note: This
     card is not used with special-purpose
     utilities.)
```

| PROGRAM | PHASE NAMES 1 THROUGH 5 | MODULE NAMES | TOS | DOS |
|---|---|---|---|---|
| Card to Disk | CDDK-CDDK2-CDDK3-CDDK4-CDDK5 | IJWCD-IJWCD1-IJWGEN-IJWCD3-IJWCD4-IJWLAB | | IJJCPD0 |
| Card to Printer and/or Punch | CDPP-CDPP2-CDPP3-CDPP4-CDPP5 | IJWCP-IJWCP1-IJWGEN-IJWCP3-IJWCP4-IJWLAB | IJJCP0 | IJJCPD0 |
| Card to Tape | CDTP-CDTP2-CDTP3-CDTP4-CDTP5 | IJWCT-IJWCT1-IJWGEN-IJWCT3-IJWCT4-IJWLAB | IJJCP0 | IJJCPD0 |
| Data Cell to Data Cell | DCDC-DCDC2-DCDC3-DCDC4-DCDC5 | IJWMM-IJWMM1-IJWGEN-IJWDD3-IJWDD4-IJWLAB | | IJJCPD0 |
| Data Cell to Disk | DCDK-DCDK2-DCDK3-DCDK4-DCDK5 | IJWMD-IJWMD1-IJWGEN-IJWDD3-IJWDD4-IJWLAB | | IJJCPD0 |
| Data Cell to Printer | DCPR-DCPR2-DCPR3-DCPR4-DCPR5 | IJWMP-IJWMP1-IJWGEN-IJWDP3-IJWDP4-IJWLAB | | IJJCPD0 |
| Data Cell to Tape | DCTP-DCTP2-DCTP3-DCTP4-DCTP5 | IJWMT-IJWMT1-IJWGEN-IJWDT3-IJWDT4-IJWLAB | | IJJCPD0 |
| Disk to Card | DKCD-DKCD2-DKCD3-DKCD4-DKCD5 | IJWDC-IJWDC1-IJWGEN-IJWDC3-IJWDC4-IJWLAB | | IJJCPD0 |
| Disk to Data Cell | DKDC-DKDC2-DKDC3-DKDC4-DKDC5 | IJWDM-IJWDM1-IJWGEN-IJWDD3-IJWDD4-IJWLAB | | IJJCPD0 |
| Disk to Disk | DKDK-DKDK2-DKDK3-DKDK4-DKDK5 | IJWDD-IJWDD1-IJWGEN-IJWDD3-IJWDD4-IJWLAB | | IJJCPD0 |
| Disk to Printer | DKPR-DKPR2-DKPR3-DKPR4-DKPR5 | IJWDP-IJWDP1-IJWGEN-IJWDP3-IJWDP4-IJWLAB | | IJJCPD0 |
| Disk to Tape | DKTP-DKTP2-DKTP3-DKTP4-DKTP5 | IJWDT-IJWDT1-IJWGEN-IJWDT3-IJWDT4-IJWLAB | | IJJCPD0 |
| Tape to Card | TPCD-TPCD2-TPCD3-TPCD4-TPCD5 | IJWTC-IJWTC1-IJWGEN-IJWTC3-IJWTC4-IJWLAB | IJJCP0 | IJJCPD0 |
| Tape to Data Cell | TPDC-TPDC2-TPDC3-TPDC4-TPDC5 | IJWTM-IJWTM1-IJWGEN-IJWTD3-IJWTD4-IJWLAB | | IJJCPD0 |
| Tape to Disk | TPDK-TPDK2-TPDK3-TPDK4-TPDK5 | IJWTD-IJWTD1-IJWGEN-IJWTD3-IJWTD4-IJWLAB | | IJJCPD0 |
| Tape to Printer | TPPR-TPPR2-TPPR3-TPPR4-TPPR5 | IJWTP-IJWTP1-IJWGEN-IJWTP3-IJWTP4-IJWLAB | IJJCP0 | IJJCPD0 |
| Tape to Tape | TPTP-TPTP2-TPTP3-TPTP4-TPTP5 | IJWTT-IJWTT1-IJWGEN-IJWTT3-IJWTT4-IJWLAB | IJJCP0 | IJJCPD0 |

Figure 3. Phase and Module Names for the File-to-File Programs

bENTRY
defines the text of the last input object module.

//bLBLTYP
- TAPE defines the reserved label area (used only if tape label checking).
- NSD(nn) used only if nonsequential files are to be copied by the copy/restore programs.

//bEXEC LNKEDT
executes the linkage editor program.

//bTLBL
tape volume-label information (used only if label checking). If running a disk program, the job control set DLBL and EXTENT must be used.

//bEXEC
defines the end-of-job control cards and signals the start of program execution.

(Utility control information is entered at this point, assuming SYSIPT and SYSRDR are assigned to the same device.)

//bEND
defines the end of utility control cards.

/&
defines the end of job.

Note: To catalog this same tape-to-tape program from the relocatable to the core image library the preceding job stream can be used with the following changes:

- //bOPTION LINK changed to //bOPTION CATAL
- Add a /& card after the //bEXEC LNKEDT card, and delete utility assignment information.

On a tape resident system the result of this job is a new resident tape that would be generated on SYS002.

After link editing the program, if you expect to rewrite the user routine or change the partition allocation, then do not delete the program modules from the relocatable library. You can then link edit the program in either the system or the private core image library again.

| PROGRAM | PHASE NAMES | MODULE NAMES | TOS | DOS |
|---|---|---|---|---|
| Assign Alternate Track-Disk | ATAD-ATAD2-ATAD3-ATAD4-ATAD5 | IJWAD-IJWAD1-IJWAD2-IJWAD3-IJWAD4-IJWAD5 | | IJJCPD1N |
| Assign Alternate Track-Data Cell | ATAM-ATAM2-ATAM3-ATAM4-ATAM5 | IJWALTM-IJWAM1-IJWAM2-IJWAM3-IJWAM4-IJWAM5 | | IJJCPD1N |
| Clear Data Cell | CLDC-CLDC2-CLDC3 | IJWCLM-IJWCLM1-IJWCLD2-IJWCLD3- | | IJJCPD0 |
| Clear Disk | CLRDSK-CLRD2-CLRD3 | IJWCLD-IJWCLD1-IJWCLD2 IJWCLD3- | | IJJCPD0 |
| Copy Disk to Card | CRDC-CRDC2 | IJWKC-IJWKC1-IJWKC2- | | IJJCPD1N |
| Copy Disk to Disk | CRDD-CRDD2 | IJWRD-IJWRD1-IJWRD2- | | IJJCPD1N |
| Copy Disk or Data Cell to Tape | CRDT-CRDT2 | IJWKT-IJWKT1-IJWKT2- | | IJJCPD1N |
| Initialize Disk | INTD-INTD2-INTD3-INTD4 | IJWID-IJWID1-IJWID2-IJWID3-IJWID4- | | IJJCPD1N |
| Initialize Tape | INTT | IJWIT | | |
| Initialize Data Cell | INTM-INTM2-INTM3-INTM4 | IJWIM-IJWIM1-IJWIM2-IJWIM3-IJWIM4 | | IJJCPD1N |
| Restore Card to Disk | CRCD | IJWRC-IJWRC1- | | IJJCPD1N |
| Restore Tape to Disk or Data Cell | CRTD | IJWRT-IJWRT1- | | IJJCPD1N |
| Tape Compare | TPCP-TPCP2-TPCP3 | IJWTPCP-IJWTCP-IJWTCP2-IJWTCP3-IJWXIT- | IJJCP0 | IJJCPD0 |
| VTOC Display Batched | LISTVTOC | IJWLVB-IJWLV1-IJWLVT- | | IJJCPD0N |
| MPS | LISTVTOC | IJWLVM-IJWLV1-IJWLVT- | | IJJCPD0N |

Figure 4. Phase and Module Names for the Special Purpose Programs

In order to link edit the utility programs to operate in a batched-jcb foreground partition in systems which do not support a private core image library, the following procedure is required for all programs except VTOC display. Place a bACTIONbFn card (where n is 1 or 2) immediately after the //bOPTION CATAL card. (For the link-edit procedure for the VTOC display program, see "VTOC Display.")

In DOS systems which support the private core image library option, the programs may be cataloged in a private core image library. A program may be link edited to execute in one or more partitions. For example:

1. Using the job stream described above, execute the linkage editor job in the background partition and catalog the program to the system core image library.

2. Using the same job stream (i.e., nc ACTION card used), execute the linkage editor job in the desired foreground partition and catalog the program to a private core image library assigned to that partition. Thereafter, when it is desired to execute the program in the foreground partition, assign the private core image library and the program will be loaded from there.

If it is desired to link edit and execute the program without cataloging to a core image library, use // OPTION LINK. However, a private core image library must be assigned when executing the linkage editor in a foreground partition. Unless stated otherwise, all further discussions of the core image library in this publication refer to the system core image library.

A typical example of the control cards used when executing a tape-to-tape program from the core image library for a distinct job is:

```
//bJOB
     user-defined job name.

//bTLBL
     tape volume and file label information
     (only if label checking).

//bASSGN
     input and output device assignments.

//bUPSI
     user-defined label processing.

//bEXEC TPTP
     program execution card.
```

(Utility control statements are entered as needed, assuming SYSIPT and SYSRDR are assigned to the same device.)

```
//bEND
     defines the end of utility control
     cards.

/&
     defines the end of job.
```

## Checkpoint Records

Checkpoint records will be handled as any other record by the assign alternate track-disk and initialize disk programs. All other utility programs ignore and bypass any checkpoint records encountered. The copy and restore disk or data cell programs provide both checkpoint and restart facilities.

# Explanations of the Utility Specifications *

## File—to—File Utilities

Seventeen utility programs are provided for the transfer of data files from any of the normal input devices to any of the normal output devices. A file can be transferred between unlike storage mediums (tape to disk), like mediums (tape to tape), or, in the case of disk to disk or data cell to data cell, the files may be transferred from one area to another area of the same unit. Throughout the general discussion of the file-to-file programs, any reference to DASD information can be equally applied to disk or data cell.

A file can be transferred from an input medium to an output medium with these options:

COPY indicates that the file is to be transferred from an input medium to an output medium without change to the format of the records or the file.

REBLOCK transfers the input file from an input medium to an output medium with only the block size being changed.

FIELD SELECT rearranges or drops fields within each input record, or converts them to zoned or packed decimal.

REBLOCK AND FIELD SELECT is a combination of the reblock and field-select options.

Printer output allows you to show the output in three ways:

DISPLAY allows you to display a byte-for-byte representation of the information.

LIST gives an edited representation of the information.

LIST AND FIELD SELECT is a combination of the list and field select options.

For the card to printer and/or punch programs, two other output options are:

BOTH PRINT AND PUNCH is a combination of copy and list.

BOTH PRINT AND PUNCH WITH FIELD SELECT combines copy and list with field select.

---

*This chapter applies only to the file-to-file utility programs, unless specifically stated otherwise.

These programs will handle fixed-length, variable-length, and undefined-length records; however, only fixed- or variable-length records can be reblocked or field selected.

If fields are selected from variable-length records, a portion of the record must be described as the fixed portion (initial section of a record that is common to all records) of the record. Only on the fixed portion can field select be employed; a field cannot be selected into the first four bytes (record length field) of the output record.

## Label Checking

The IBM System/360 Disk and Tape Operating Systems utility programs process tape and DASD labels like the Disk and Tape Operating Systems IOCS. For information on label checking see the Supervisor and I/O Macros publications as listed in the preface of this publication.

NONSTANDARD AND USER LABEL HANDLING

Tape files containing IBM standard labels or no labels at all can be processed without providing a user routine. When using a file-to-file utility program to process nonstandard or user labels or when no label checking is desired, an UPSI job control card is required. The first five bits (bits 0-4) of the UPSI byte have been assigned as shown below. The low-order bits (bits 5-7) have no significant values for the file-to-file utilities. (If an UPSI byte is significant to a special-purpose utility, it is explained within the individual program description.)

Left to right, bits 0-4 of the UPSI byte must be set as follows (0 equals off, 1 equals on):

Bits 0 and 1 are switches for input label checking.

Bit 0 Off = standard input-label checking.
On = nonstandard or no input-label checking.

Bit 1 Off = no user input-label checking.
On = user input-label checking.

Bits 2 and 3 are switches for output label checking.

Bit 2 Off = standard output-label checking.
On = nonstandard or no output-label checking.

Bit 3 Off = no user output-label checking.
On = user output-label checking.

Bit 4 is a switch for nonstandard or no output-label handling.

Bit 4 Off = write leading tapemark.
On = do not write leading tapemark.

Bit 7 is used for forced-end-of-volume (FEOV).

Bit 7 Off = the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).
On = the sequential file to be copied from disk or data cell to printer, punch, disk or tape supports FEOV.

When an UPSI card is supplied to a program, the byte is propagated from job step to job step unless another UPSI card is supplied to reset the bits. All of the UPSI bits are set to 0 following each job performed unless a new statement is supplied. When rightmost bits are not set by an UPSI statement, they are assumed to be zero.

Examples:

An UPSI card is required for no input-label and standard output-label checking:

```
//bUPSI 10000000
```

An UPSI card is required for no input-label or output-label checking with a leading tapemark on the output:

```
//bUPSI 10110000
```

An UPSI card is not required when there is standard input-label, output-label, and no user label checking.

A user label routine must be supplied only if bits 1 or 3 of the UPSI byte are on. You must supply your label checking routine in assembled, relocatable format. This control section must define three symbolic names as entry (ENTRY) points:

IJWLABIN The symbolic entry point to the input-label processing section of the user's routine.

IJWLABOU The symbolic entry point to the output-label processing section of the user's routine.

IJWLABND The symbolic entry to represent the last location +1 of the program.

After the program is loaded, control is given to your initialization routine through the address found in the END card (assembly program END card). You can then perform any initialization desired before label checking. Upon completion of initialization, you must branch back to the utility program (the return address is found in register 14). The initialization routine may consist of only the return branch instruction. All other entries made to the user's routine will be made through the symbolic names IJWLABIN or IJWLABOU. To return from IJWLABIN and IJWLABOU user-label processing to IOCS label processing, use the LBRET macro instruction. (See the Supervisor and I/O Macros publications listed in the preface of this publication.) The user's routine will be entered from the IOCS label-processing routines.

The user's routine can be assembled with a 10K assembler. This routine has access to all IOCS macros except those which use the transient area (CANCEL, EOJ, OPEN, CLOSE).

For further information concerning communication with the IOCS open and close routine, see the Supervisor and I/O Macros publications listed in the preface of this publication.

## Utility Message Routines

The message routine of the utility programs is available for your use. The entry point to the message routine is located at the symbolic address, IJWxxxMS (xxx can be determined from Figure 5). Your user routine may not have access to register 4 and must supply registers 0, 1, and 7 with the following information:

Reg 0 The length of the message.

Reg 1 The address of the first byte of the message.

Reg 7 The return address to the user's routine.

No diagnostics will be performed on the contents of these registers.

If the first character of a message is nonblank, the message is printed on SYSLST and SYSLOG and a reply is requested from

SYSLOG. The reply, or answer byte given, must be one character located at the symbolic address IJWxxxAN on return from the message routine (xxx can be determined from Figure 5). If the first character of a message is blank, the message is printed only on SYSLST. In either case, the first character of the message is not printed.

If a message is printed that requires a reply and SYSLOG is a printer, X'FF' will be in the answer byte (IJWxxxAN) on return from the message routine.

If SYSLST and SYSLOG are the same printer and the message was designated to both, the message will only appear once.

| xxx | Meaning |
|---|---|
| CD1 | Card to Disk Program |
| CT1 | Card to Tape Program |
| DC1 | Disk to Card Program |
| DD1 | Disk to Disk Program |
| DM1 | Disk to Data Cell Program |
| DP1 | Disk to Printer Program |
| DT1 | Disk to Tape Program |
| MD1 | Data Cell to Disk Program |
| MM1 | Data Cell to Data Cell Program |
| MP1 | Data Cell to Printer Program |
| MT1 | Data Cell to Tape Program |
| TC1 | Tape to Card Program |
| TD1 | Tape to Disk Program |
| TM1 | Tape to Data Cell Program |
| TP1 | Tape to Printer Program |
| TT1 | Tape to Tape Program |
| TCP | Tape Compare Program |

Figure 5. Answer Byte or Entry Point Completions

## Multifile Volume (Tape)

The utility programs may be used to build multifile volumes and read from them at later dates. File positioning will be performed by logical IOCS only if the files are labeled with IBM standard labels. Output files, nonstandard labeled files and unlabeled files are not acceptable. The filename, volume-sequence, and file-sequence numbers must be placed in the TLBL card. The positioning performed must be by the use of the magnetic tape command (MTC).

Reference information on the MTC command can be found in the System Control and Service publications as listed on the front cover of this publication.

When using the utility programs to process multifile tape input volumes, the no-rewind-option (IN) parameter found in the utility modifier statement must be specified.

## Multivolume Files (Tape)

Input or output files to these programs can consist of multiple volumes. The multiple volumes must belong to the same data files. The control statement entries used to process the first volume are used to process each successive volume because the same fields are checked in each volume. Each tape reel of a multivolume tape file is unconditionally rewound and unloaded if no alternate tape drive has been assigned. In all other cases the volume will be treated as specified by the input or output parameter in the utility modifier statement.

When alternate tape drives are specified and processing is completed on a particular file, the last drive processed will become the primary drive. If a new job is executed at this time, the last drive processed will then become the primary drive unless a reassignment of tape drives is made.

## Record Skipping

Any number of logical records (up to 99,999,999) may be bypassed before processing is performed. This number can be indicated in the Rx parameter of the utility modifier statement. The number indicated in the parameter will be the first record to be processed.

Record skipping cannot be performed for the copy function (TC), and if specified, it will be ignored. To skip records at the beginning of a file and copy the remainder, the reblock function (TR) must be indicated, and the input-description and output-description parameters must contain identical values.

## Sequence Numbering

Sequence generation on card output can be indicated in the utility modifier statement. A field up to ten characters long can be punched into each card. This field is numbered starting from 1 (witn high-order zeros) and is increased by 1 for each succeeding card. If a sufficiently long field is not defined to number all of the cards, the number wraps around to zerc

without an error indication. The sequence number overlays any data selected into the sequence area of the card. Sequence checking also can be performed for card input to assure ascending sequence of the specified field. If a card is out of sequence, a message is written on SYSLST and processing continues.

## Binary Records

When processing cards punched in column binary format, the input and output parameters (parameters A and B in the utility modifier statement) must be specified as twice the number of card columns used. Since the column binary special feature on the card read-punch separates the data in each card column punched in binary into one odd and one even byte, a buffer size of 160 bytes is required for an 80-byte binary record.

## Printer Output

Printer output can be in 120-, 132-, or 144-character line length depending on the printer being used and the output format. Printer output can be in either data display or list format. Examples of these formats appear in Appendix B. Printer Output.

DATA DISPLAY

The data display format provides a visual picture of the data file. Fixed, variable and undefined records can be handled, but the field select option cannot be used. Every byte of data in the file appears in the printed output. If data display is specified (TD), 120-character line is forced. Only portions of the print line are used for data. The first twenty positions (columns 1-20) are reserved for information describing the file, such as block size, block number, and record number. Data is normally displayed in hexadecimal form but may optionally be displayed in alphameric form. A heading line can be printed. A scale line is printed at the top and bottom of each page. If record length is specified as fixed-length or variable-length, each logical record starts on a new line. The input block size is printed only if the input length is not equal to the specified block size. The excess is not printed when the specified maximum length block size is exceeded. Single spacing is used between lines of print.

DATA LIST

The data list format provides a simple edited listing of the file. The entire print line is available for data output; output is restricted to one line per logical record. The input record length cannot exceed the size of the print line. If so, field select is required. Either 120-, 132-, or 144-character line length can be specified in the utility modifier statement. Fields can be selected to unpack, convert to hexadecimal representation, and format the page. Data list mode allows only character printing unless a hexadecimal field is selected through a field select entry.

Page numbers normally print at the bottom of each page but may be suppressed. A heading line can be optionally printed.

## Available I/O Area

Each of these utility programs can run in a 10K partition. The maximum amount of storage available as I/O area is the area beginning at the end of the program being run and extending to the end of the available storage. The available storage area is reduced by:

- Field-selection, which is generated in upper storage. The instructions necessary to move and process each field defined reduce the available I/O area.
- Reblocking, which is generated in upper storage. The I/O area is reduced by the number of instructions necessary to move one record.
- Supervisor, which can immediately preceed the origin location of the utility program. A large supervisor, therefore, reduces the I/O area.

MINIMUM I/O AREA

Before reduction of the I/O area, caused by the type of user processing to be performed, the programs ensure you of these minimum I/O areas (main storage sizes are based on a supervisor of 6,144 bytes):

File-to-File Utilities

| | |
|---|---|
| Card to Disk | Not less than 6,000 bytes. |
| Card to Printer and/or Punch | Not less than 4,500 bytes. |

| | |
|---|---|
| Card to Tape | Not less than 6,400 bytes. |
| Data Cell to Data Cell | Not less than 5,500 bytes. |
| Data Cell to Disk | Not less than 5,500 bytes. |
| Data Cell to Printer | Not less than 5,000 bytes. |
| Data Cell to Tape | Not less than 5,800 bytes. |
| Disk to Card | Not less than 5,200 bytes. |
| Disk to Data Cell | Not less than 5,500 bytes. |
| Disk to Disk | Not less than 5,500 bytes. |
| Disk to Printer | Not less than 5,000 bytes. |
| Disk to Tape | Not less than 5,800 bytes. |
| Tape to Card | Not less than 6,000 bytes. |
| Tape to Data Cell | Not less than 5,400 bytes. |
| Tape to Disk | Not less than 5,400 bytes. |
| Tape to Printer | Not less than 5,100 bytes. |
| Tape to Tape | Not less than 5,900 bytes. |

Special-Purpose Utilities

| | |
|---|---|
| Assign Alternate Track-Data Cell | Enough to assign one track at one time. |
| Assign Alternate Track-Disk | Enough to assign one track at one time. |
| Clear Data Cell | Enough to clear three tracks at one time. |
| Clear Disk | Enough to clear one track at one time. |
| Copy-Disk to Disk | Not less than 6,400 bytes. |
| Copy and Restore-Disk to Card | Not less than 6,400 bytes. |
| Copy and Restore-Disk or Data Cell to Tape | Not less than 6,800 bytes. |
| Initialize Data Cell | Enough to initialize one track at one time. |
| Initialize Disk | Enough to initialize one track at one time. |
| Tape Compare | Not less than 6,000 bytes. |
| VTOC Display | 140 bytes; more I/O area will not be used. |

## I/O Area Assignment

If the utility program can assign two input or output areas, overlap of the I/O operations can be performed whenever channel assignment permits. The utility program determines the method of I/O area assignment based on the maximum block size, the available I/O area, and the type of job being processed.

For the copy and both print and punch functions, the I/O area assignments may be:

2 input/output areas
1 input/output area

For the field select, reblock, reblock and field select, list, list and field select, data display, and both print and punch with the field select function, the I/O area assignments may be:

2 input and 2 output areas
1 input and 2 output areas
2 input and 1 output areas
1 input and 1 output area

## First—Character Forms Control

When the first-character forms control is used, the first character of the data record is considered to be the forms-control character and is printed unless excluded by field selection. For fixed-length records the forms-control

character is the first character of the logical record. For variable-length records the forms-control character is the first character following the record-length field. First-character forms control is invalid for records with key fields or data display. This option allows a choice of four standards by which forms control can be regulated:

S {A, B, C, D}

The type of forms-control character to be recognized can be indicated in the S parameter of the tape-, disk-, and data cell-to-printer programs.

TYPE A

Type A indication allows you to use the character that is the command-code portion of the System/360 channel command word used in printing a line or spacing the forms. If the character is not one of the following characters, single spacing after printing is performed and no error indication is given. Printing occurs only for command codes which include a print in the operation.

| 8-Bit Code | Punch Combination | Function |
|---|---|---|
| 00000001 | 12,9,1 | Write (no automatic space) |
| 00001001 | 12,9,8,1 | Write and space 1 line after printing |
| 00010001 | 11,9,1 | Write and space 2 lines after printing |
| 00011001 | 11,9,8,1 | Write and space 3 lines after printing |
| 10001001 | 12,0,9 | Write and skip to channel 1 after printing |
| 10010001 | 12,11,1 | Write and skip to channel 2 after printing |
| 10011001 | 12,11,9 | Write and skip to channel 3 after printing |
| 10100001 | 11,0,1 | Write and skip to channel 4 after printing |
| 10101001 | 11,0,9 | Write and skip to channel 5 after printing |
| 10110001 | 12,11,0,1 | Write and skip to channel 6 after printing |
| 10111001 | 12,11,0,9 | Write and skip to channel 7 after printing |
| 11000001 | 12,1 | Write and skip to channel 8 after printing |
| 11001001 | 12,9 | Write and skip to channel 9 after printing |
| 11010001 | 11,1 | Write and skip to channel 10 after printing |
| 11011001 | 11,9 | Write and skip to channel 11 after printing |
| 11100001 | 11,0,9,1 | Write and skip to channel 12 after printing |
| 00001011 | 12,9,8,3 | Space 1 line immediately |
| 00010011 | 11,9,3 | Space 2 lines immediately |
| 00011011 | 11,9,8,3 | Space 3 lines immediately |
| 10001011 | 12,0,8,3 | Skip to channel 1 immediately |
| 10010011 | 12,11,3 | Skip to channel 2 immediately |
| 10011011 | 12,11,8,3 | Skip to channel 3 immediately |
| 10100011 | 11,0,3 | Skip to channel 4 immediately |
| 10101011 | 11,0,8,3 | Skip to channel 5 immediately |
| 10110011 | 12,11,0,3 | Skip to channel 6 immediately |
| 10111011 | 12,11,0,8,3 | Skip to channel 7 immediately |

| | |
|---|---|
| 11000011 12,3 | Skip to channel 8 immediately |
| 11001011 12,0,9,8,3 | Skip to channel 9 immediately |
| 11010011 11,3 | Skip to channel 10 immediately |
| 11011011 12,11,9,8,3 | Skip to channel 11 immediately |
| 11100011 0,3 | Skip to channel 12 immediately |
| 00000011 12,9,3 | No op |

TYPE B

Type B allows you to use the d-modifier character of the IBM 1401 carriage-control instruction used in printing a line or spacing forms with a 1401 system. Printing occurs only for the d-modifiers which include a print in the operation. If the character read is not one of the valid characters, the line will be printed after single spacing and no error indication will be given. The codes are:

| d | Immediate Skip To | d | Skip After Print To |
|---|---|---|---|
| 1 | Channel 1 | A | Channel 1 |
| 2 | Channel 2 | B | Channel 2 |
| 3 | Channel 3 | C | Channel 3 |
| 4 | Channel 4 | D | Channel 4 |
| 5 | Channel 5 | E | Channel 5 |
| 6 | Channel 6 | F | Channel 6 |
| 7 | Channel 7 | G | Channel 7 |
| 8 | Channel 8 | H | Channel 8 |
| 9 | Channel 9 | I | Channel 9 |
| 0 | Channel 10 | ? | Channel 10 (EBCDIC or BCDIC) |
| # | Channel 11 | . | Channel 11 |
| @ | Channel 12 | ¤ | Channel 12 (EBCDIC or BCDIC) |

| d | Immediate Space | d | After Print Space |
|---|---|---|---|
| J | 1 space | / | 1 space |
| K | 2 spaces | S | 2 spaces |
| L | 3 spaces | T | 3 spaces |

TYPE C

Type C allows the use of the following codes as forms-control characters. If the character read is not one of the valid characters, the line will be printed with single spacing after printing and no error indication will be given.

| Code | Space or Skip Action |
|---|---|
| + (EBCDIC or BCDIC) | Suppress space and print. |
| Blank | Print and single space. |
| 0 | Double space, print, and space. |
| - | Triple space, print, and space. |
| 1-9 or J-R | Immediate skip to channel 1-9 (i.e., 1 or J=skip to channel 1; 2 or K=skip to channel 2, etc), print and then space. |

TYPE D

Type D allows the use of the ASA FORTRAN forms-control characters. If the character read is not one of the valid characters, the line will be printed with single spacing before printing and no error indication will be given.

| Code | Space or Skip Action |
|---|---|
| Blank | Space one line before printing. |
| 0 | Space two lines before printing. |
| - | Space three lines before printing. |
| + (EBCDIC or BCDIC) | Suppress space before printing. |
| 1 | Skip to channel 1 before printing. |
| 2 | Skip to channel 2 before printing. |
| 3 | Skip to channel 3 before printing. |
| 4 | Skip to channel 4 before printing. |
| 5 | Skip to channel 5 before printing. |
| 6 | Skip to channel 6 before printing. |

| | |
|---|---|
| 7 | Skip to channel 7 before printing. |
| 8 | Skip to channel 8 before printing. |
| 9 | Skip to channel 9 before printing. |
| A | Skip to channel 10 before printing. |
| B | Skip to channel 11 before printing. |
| C | Skip to channel 12 before printing. |

SYSLST/SYS005 CARRIAGE CONTRCL

When separate printers are assigned to SYSLST and SYS005 or the same device is assigned to both, the controlling factcr in carriage control skipping must be determined. The possible printer assignment and the determining carriage control factors are:

| Printer Assignment | Control Factors |
|---|---|
| SYSLST as a separate printer. | LINECT (line count) operand in the set command. |
| SYS005 as a separate printer. | Sensing either channel 12 or the proper forms-control character. |
| SYSLST and SYS005 as the same printer. | Forms-control charac-ter or, if none is present, the LINECT operand in the set command. Channel 12 will not be detected. |

## First—Character Stacker—Select Control

First-character stacker-select control can be specified for the tape and disk to card programs. The stacker-select control character must be the first character cf the data portion of the record and is punched unless excluded by field select. These characters cause the indicated action. Any other character will cause the selection of pocket 1.

| Character | Action |
|---|---|
| V | Select pocket 1 |
| W | Select pocket 2 |

## Utility Modifier Statement

This statement is used with the file-to-file programs and allows you to describe the input file that is to be processed and the output file that is desired. If the statement is present and cptional parameters are left out, default values are assumed.

When a file is to be copied without change, it is possible to use the program without the presence of a utility modifier. All record statement formats (fixed-length, variable-length, undefined) may be copied without change as long as maximum block sizes do not exceed the default values of the particular program. If default values are exceeded, the output block is truncated.

The values the program assumes are unique to each program and are given in the discussion of each program.

The general format of the utility modifier statement is:

```
//bUxxbTt,Ff,A=(input),B=(output),E=(c,d),
  Ix,Ox,Px,Q=(x,y),Rx,Sx
```

Figure 6 shows detailed information of the entries in the utility modifier statement.

```
//bUxxb
```

`//bU`
identifies this as a utility modifier statement. (The letter b always indicates a blank space.)

`xxb`
these are the initials of the program and can be omitted if this statement is to be used with more than one program.

Following these identifiers the desired parameters are indicated. Each parameter must be followed by a comma, except the last parameter which must be followed by at least one blank. The optional parameters [E=(c,d),Ix,Ox,Px,Q=(x,y),Rx,Sx] can be omitted from the utility modifier statement and assumed values made. Commas should not be entered to indicate omitted parameters.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF<br>TD<br>TL<br>TLF<br>TB<br>TBF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| | | D | Display |
| | | L | List |
| | | LF | List and Field Select |
| | | B | Both Print and Punch |
| | | BF | Both Print and Punch with Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable-length input records, the fixed portion of each input record (the letter n) and the maximum block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| | A=(K=l,D=l) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=l,D=l) | For fixed-length DASD input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two entries must be separated by a comma and enclosed in parentheses. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable-length input records without field select, the maximum block length must be enclosed in parentheses. |
| Output Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m). | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable-length output records, the fixed portion of each output record (the letter n) and the maximum output block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| | B=(K=l,D=l) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (K=l,D=l) | For fixed-length DASD output records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two entries must be separated by a comma and enclosed in parentheses. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |

Figure 6. Utility Modifier Statement Parameters (Part 1 of 2)

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| | | (g) | For undefined output records or variable-length output records without field select, the maximum block length must be enclosed in parentheses. |
| | B=(p) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (p) | For printer output the size of the print line (120, 132, 144) must be entered. |
| | B=(n,p) | B= | This letter and symbol indicate this is the output description parameter. |
| | | (n,p) | For field select of variable-length records with printer output records, the fixed portion of each output record (the letter n) and the size of the print line(the letter p) must be enclosed in parentheses and separated by a comma. |
| Optional | Ix<br>Ox<br>Sx<br>Px<br>Rx<br>Q=(x, y)<br>E=(x, y) | | These parameters are unique to each program and are explained under the discussions of the individual programs. |

Figure 6. Utility Modifier Statement Parameters (Part 2 of 2)

The first parameter, indicated by Tt in the general format, describes the type of function to be performed. This parameter is required in all utility modifier statements. The letter T is entered to identify this parameter and is followed by one or two additional letters to indicate the type of function to be performed:

TC Copy
TF Field select
TR Reblock
TRF Reblock and field select.

The parameters for printer output programs are:

TD Data display (a byte-for-byte representation of the file)

TL List (an edited representation of the file)

TLF List and field select.

The parameters for printed and punched output with the card-to-printer and/or punch program are:

TB Both print and punch

TBF Both print and punch with field select.

The second parameter, indicated by Ff in the general format, describes the format of the records to be processed for input and output. This parameter is required in all utility modifier statements. The letter F is entered to identify this parameter and is followed by an additional letter to indicate the exact record format:

FF Fixed-length records
FV Variable-length records
FU Undefined-length records.

The third parameter indicated by A=(input) in the general format, is the input-file description. This parameter is required in all utility modifier statements and is entered in one of three forms:

| Format | Maximum Number of Digits |
|---|---|
| A=(n,m) | 4,5 |
| A=(K=l,D=l) | 4,5 |
| A=(g) | 5 |

A=(n,m)
is indicated for fixed-length records without key fields and variable-length input records with field select without key fields. The letter A and symbol = identify this as the input-file description. The (n,m) indicates that the input record length or the fixed portion of variable-length records (the letter n) and input block length or maximum block length (the letter m) for variable-length records should be entered, separated by a comma, and enclosed in parentheses. If a fixed input is 50 characters long and the block length is 250 characters long, the input parameter must be entered A=(50,250) and must be followed by a

comma to separate this parameter from the one following.

A=(K=l,D=l)
is indicated for fixed-length DASD records when key fields are present. The letter A and symbol = identify this as the input-file description. The (K=l,D=l) indicates that the letter K and symbol = are followed by the length of the key and that the letter D and symbol = are followed by the length of the data field. These must be separated by a comma and enclosed within parentheses. If a DASD input record has a key length of 10 and data field length of 60, the input parameter must be entered A=(K=10,D=60) and must be followed by a comma to separate this parameter from the one following.

A=(g)
is indicated for undefined input records and variable-length input records without field select. The letter A and the symbol = identify this as the input-file description. The (g) indicates that the maximum input block length is to be entered in parentheses. If a file of undefined records contains a maximum block length of 300, the input parameter must be entered A=(300) and must be followed by a comma to separate this parameter from the one following.

The fourth parameter, indicated by B=(output) in the general format, is the output-file description, and is entered in one of five forms:

| Format | Maximum Number of Digits |
|---|---|
| B=(n,m) | 4,5 |
| B=(K=l,D=l) | 4,5 |
| B=(g) | 5 |
| B=(p) | 3 |
| B=(n,p) | 4,3 |

B=(n,m)
is indicated for fixed-length records without key fields and variable-length records with field select without key fields. The letter B and the symbol = identify this as the output-file description. The (n,m) indicates that the output record length or the fixed portion of variable-length records (the letter n) and the output block length or maximum block length (the letter m) for variable-length records should be entered, separated by a comma and enclosed in parentheses. If a fixed-length output record length is 50 characters long and the block length is 250 characters long, the output parameter must be entered B=(50,250) and must be followed by a comma if another parameter is to follow.

B=(K=l,D=l)
is indicated for fixed-length DASD records when key fields are present. The letter B and symbol = identify this as the output-file description. The (K=l,D=l) indicates that the letter K and symbol = are followed by the length of the key, and the letter D and symbol = are followed by the length of the data field. These must be separated by a comma and enclosed within parentheses. If a DASD output record has a key length of 10 and a data field length of 60, the output parameter must be entered B=(K=10,D=60) and must be followed by a comma if another parameter is to follow.

B=(g)
is indicated for undefined output records and variable-length output records without field select. The letter B and the symbol = identify this as the output-file description. The (g) indicates that the maximum output block length is to be entered within parentheses. If an output file of undefined records is to contain a maximum block length of 300, the output parameter must be entered B=(300) and must be followed by a comma if another parameter is to follow.

B=(p)
is indicated for printer output programs. The letter B and the symbol = identify this as the output-file description. The (p) indicates the size of the print line (120, 132, or 144).

B=(n,p)
is indicated for printer output programs with field select of variable-length records. The letter n indicates the last print position that may be used for field selection. If copy variable is to be performed, the variable portion of the record will follow the nth print position. The last print position (the letter n) and the size of the print line (the letter p) must be enclosed in parentheses and separated by a comma.

PARAMETER COMBINATIONS

The record format, input-file description, and output-file description parameters can be presented in these possible forms:

```
FF,A=(n,m),B=(n,m)
FF,A=(K=l,D=l),B=(n,m)
FF,A=(n,m),B=(K=l,D=l)
FF,A=(K=l,D=l),B=(K=l,D=l)
FV,A=(n,m),B=(n,m)
FV,A=(g),B=(g)
FU,A=(g),B=(g)
```

Note: The optional parameters [E=(c,d),Ix,Ox,Px,Q=(x,y),Rx, and Sx] are unique to each program and are explained under the discussions of the individual programs.

For printer output there are five additional forms:

```
FF,A=(n,m),B=(p)
FF,A=(K=l,D=l),B=(p)
FV,A=(g),B=(p)
FU,A=(g),B=(p)
FV,A=(n,m),B=(n,p)
```

FIELD SELECT STATEMENT

With this option a field in each input record or the fixed portion of variable-length records can be moved to a different relative location in the corresponding output record. Those areas of the output record that are not filled with selected fields are filled with blanks (X'40'). A selected field can be moved in the following ways:

- Moved without change.
- Moved and converted from zoned to packed decimal.
- Moved and converted from packed to zoned decimal.
- Moved and converted to hexadecimal for printer output.

Converting a field causes the output field to be smaller or larger than the input field. A field converted to hexadecimal representation for printer output requires twice the amount of area as that required for input.

When field select is used, only those bytes in the input record that are selected will be transferred to the output record. It is therefore possible with field select to ignore certain fields and have them dropped from the output record. The section of a variable-length record that is not defined as the fixed portion can be copied onto the output record. As a result of dropping fields or changing field formats, it is possible to have output records of a different length from the input records.

The utility programs generate the necessary instructions for this option. This technique provides optimum performance for the user.

A key field can be selected from or placed into the key portion of a DASD record. The field that is selected must be completely contained within the key field or data field. A field that is placed in a key field or data field must be placed entirely in the key portion or the data portion of the record. Fields are selected or placed, relative to one of the first byte (i.e. byte 1, not byte 0) of either the key or data field.

The field select control statement provides the information for the file-to-file programs to transfer fields from an input record to the same or a different relative location of the output record. If the field select statements are punched in cards, each card need not be filled even if additional field select cards follow. Each card should begin with //bFSb. As many field select statements as necessary may be used. For example:

```
//bFSb1,15,1/16,(P,5,3),16/72,(P,3,2),
   19/75,(P,6,4),21/23,(P,8,5),25/

//bFSb21,2,30
```

The field selected must be complete on one statement. Field select can be performed on any portion of fixed-length records. Only fields within the fixed portion of each variable-length record can be selected.

The fixed-length portion of a variable-length record is the initial section of a record that is common to all records. The first four bytes of this fixed portion are always the record-length indication.

For nonprinter programs involving variable-length records, the record length is generated into the first four bytes of each output record. The generation of this field prohibits field selection from being performed in this area. When performing field selection with nonprinter, variable- or fixed-length records, the r and t in the field selection parameter (r,s,t) are relative to the first byte of the record, which includes the four-byte record length indication.

For printer programs (list mode) involving variable-length records, the record length is not generated into the output record unless field selected. When printer output field selecting of variable-length records is performed, the r in the field selection parameter (r,s,t) is relative to the first byte of the record including the four-byte record length indication, and the t is relative to the first print position of the print line. The remainder of the variable-length record can be copied onto the output record if indicated in the field select statement. The format and contents of this statement are:

//bFSbr,s,t/r,s,t/r,s,t

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (see note below), of the field in the input record to be selected. For binary records, this number is relative to the record as it appears in main storage, not on the statement.<br><br>,(comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/(slash) separates selected fields. |

Note: Bytes of a record are numbered consecutively beginning from byte 1, not 0.

When a field is to be selected from a key field (DASD input), the letter K followed by a comma and the starting position of the field to be selected must be placed in parentheses.

Example: //bFSb(K,r),s,t

When a field is to be placed into a key field (DASD output), the letter K followed by a comma and the starting position of the field in the output record must be placed in parentheses.

Example: //bFSbr,s,(K,t)

When a field is to be selected from a key field (DASD input) and is to be placed into a key field (DASD output), the starting position of the field in the input record and output record must be preceded by the letter K and a comma, and enclosed in parentheses.

Example: //bFSb(K,r),s,(K,t)

The other operations (pack, unpack, and hexadecimal) are defined in the field-length portion of the parameter. These operations are independent of whether the field source or destination is a key.

PACK

When the input field is to be packed before it is placed in the output record (invalid for printer output), the field select parameter will appear in this form:

r,(P,n,m),t

P
: identifies the pack operation.

n
: is the size of the input field.

m
: is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
: identifies the unpack operation.

n
: is the size of the input field.

m
: is the size of the output field.

A packed field contains the sign in the right four bits of the low-order byte. When the field is unpacked, the sign is contained in the left four bits of the low-order byte. The input record length, denoted by (n), is in the field select statement //bFSbr,(U,n,m),t. If (n) is made the same length as the packed decimal field, the last printed character will indicate the numeric value of the last digit and the sign of the field. For example, 123C will be printed 12C. To

print the value of the last numeric digit and drop the sign, you must increase the input record length by one.

//bFSb1,(U,3,5)1

Record is 123C, unpacked F1F2C3, printed 12C.

//bFSb1,(U,4,5)1

Record is 123CD0, unpacked F1F2F3FC0D, printed 123.

HEXADECIMAL

When a program has printed output, the field selected may be printed in hexadecimal representation. This operation is indicated as follows:

r,(X,n),t

X
identifies the hexadecimal operation;

n
is the size of the input field.

Only the field length of the input is necessary for this operation because the output length will always be assumed to be twice as large. X and n are enclosed in parentheses and separated by a comma.

COPY VARIABLE

When the section of a variable-length record not defined as the fixed portion is to be copied, the letters CV (copy variable) must be present in the field select control statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered as one of the following:

- Before the first field to be selected (e.g., //bFSbCV/r,s,t).
- Between selected fields (e.g., //bFSbr,s,t/CV/r,s,t).
- Following selected fields (e.g., //bFSbr,s,t/r,s,t/CV).

The variable section of the record is placed in the output record following the fixed portion of the record as defined in the output description parameter.

HEADING STATEMENT

A heading line can be printed for programs with printed output. Heading lines are ignored if first-character forms control is specified. A maximum of two statements can be used to indicate the heading line desired, but the second statement need not be entered if the first statement contains all of the desired information. The first statement is entered //bH1b (followed by the information to be printed in print positions 1-74). The second statement is entered //bH2b (followed by the information to be printed in the rest of the heading line). Heading statements must follow any field select statements used.

Example: //bH1bLISTbOFbDATAbCELLbFILE

## END Statement

This statement must be the last of the utility control statements in the program.

//bEND
//b
indicates that this is a control statement.

END
indicates the last utility modifier control statement.

## Examples

The following are examples from a payroll file of utility modifier statement and field select statement preparation (one card-to-tape, two tape-to-tape, and one disk-to-disk) for creating a file of fixed-length records for testing.

CARD TO TAPE

The input file contains eight fields. The fields numbered 1, 2, 7, 8, 4, and 3 are to be moved, in that order, to the output area and fields 2, 4, 7, and 8 are to be packed while being moved.

1. Name in positions 1-15.
2. Hourly rate in positions 16-20.
3. Number of dependents in positions 21-22.
4. Earnings to date in positions 23-30.
5. Address in positions 31-66.
6. Date of service in positions 67-71.
7. Hours worked in positions 72-74.
8. Weekly earnings in positions 75-80.

The utility modifier statement is entered as:

```
//bUCTbTF,FF,A=(80,80),B=(80,80)
```

The field select statement is entered as:

```
//bFSb1,15,1/16,(P,5,3),16/72,(P,3,2),
   19/75,(P,6,4),21/23,(P,8,5),25/21,2,30
```

TAPE TO TAPE

If an exact copy is to be made of the input file, a field select statement is not needed. The input-file format is the same as the card-to-tape program.

The utility modifier statement is entered as:

```
//bUTTbTC,FF,A=(80,80),B=(80,80)
```

An alternate method occurs when the input file contains variable-length records. The minimum length logical record is 24 bytes, and the maximum block length is 300 bytes. The fixed portion of the logical record is defined as 24 bytes and consists of two ten-byte fields and the record-length field. The two ten-byte fields are to be interchanged, and the variable portion of each logical record is to be copied.

The utility modifier statement is entered as:

```
//bUTTbTF,FV,A=(24,300),B=(24,300)
```

The field select statement is entered as:

```
//bFSb5,10,15/15,10,5/CV
```

DISK TO DISK

The input file contains nine data fields and a key field. The first field (1) is the key field and is to be transferred to the output key field. Field 2 is to be dropped. Fields 3, 4, 9, 10, 6, and 5, in that order, are to be transferred to the output record. Fields 4, 5, 6, 9, and 10 are to be packed while being moved.

1. Man number in positions 1-10 of the ten-position key field.
2. Department number in positions 1-5.
3. Name in positions 6-20.
4. Hourly rate in positions 21-25.
5. Number of dependents in positions 26-27.
6. Earnings to date in positions 28-35.
7. Address in positions 37-71.
8. Date of service in positions 72-76.
9. Hours worked in positions 77-79.
10. Weekly earnings in positions 8-85.
11. Positions 86-100 unused.

The utility modifier statement is entered as:

```
//bUDDbTF,FF,A=(K=10,D=100),B=(K=10,D=31)
```

The field select statement is entered as:

```
//bFSb(K,1),10,(K,1)/6,15,1/21,(P,5,3),16
```

## Key Fields

DASD processing begins in the area of DASD indicated in the EXTENT statement as the lower limit, and continues consecutively until the upper limit or EOF is reached. A field can be selected from, or placed into, the key portion of a DASD record. The field that is selected must be completely contained within the key field or data field. A field that is placed in a key field or a data field must be placed entirely in the key portion or data portion of the record. DASD files without keys are handled without consideration to the key field and can be thought of as being similar to tape files.

Disk files with key fields require information unique to key fields processing. The records must be fixed length and unblocked or one of the following types of records identified as an undefined record:

Fixed-length blocked
Variable-length blocked or unblocked
Undefined.

The records identified as undefined records with keys are restricted to being copied or displayed and are valid for DASD-to-DASD and DASD-to-printer programs only.

DASD FILES WITH KEY FIELDS (FIXED-LENGTH UNBLOCKED)

Key fields are only valid for:

- DASD output.
- DASD input and DASD output.

- DASD input and printer output (printer output is capable of printing key fields).

DASD to Card or Tape

To transfer data from DASD to card or tape, field select must be used to transfer the key field to a data field for output. Depending upon the output desired, certain information is required.

For tape output:

1. Field select must be used.
2. Reblocking and field select together can be specified for blocked output records.

For card output:

1. Field select must be used.
2. Reblocking and field select together are not valid because disk input is unblocked and card output must be unblocked.

Card or Tape to DASD

When data is transferred from card or tape to DASD, field select must be used to create the key field for output. Depending upon the output desired, certain information is required.

For card input:

1. Field select must be used.
2. Reblocking and field select together are not valid, because both card input and disk output must be unblocked.

For tape input:

1. Field select must be used.
2. Reblocking and field select together must be specified when the input is blocked.

DASD to Printer

When a DASD file is printed, it is possible to print the key fields by either the display or list print format.

Display — The key field must be specified on the utility modifier card in the format (K=1,D=1). This will cause both the key and data field to be printed.

List — Field select can be used to select a field from the key for printing. If field select is not used, the key and data must fit on the print line.

DASD to DASD

When records from DASD to DASD are transferred with these key field conditions, the following functions can be performed:

Copy — The file is transferred without change.

Field select — The file can be transferred with:
Data fields dropped or rearranged,
Record length changed,
Key fields changed.

Key Fields on Input and No Key Fields on Output:

Field select — Field select must be used to:
Remove the key field from the data,
Remove the key field and drop or rearrange data fields,
Remove the key field and change the record length.

Reblock & field select — This function can be used to do those options under field select and provide blocked output records.

No Key Fields on Input and Key Fields on Output (Unblocked Input):

Field select — Field select must be used to:

Create key fields,
Create key fields and drop or rearrange data fields,
Create key fields and change the record length.

No Key Fields on Input and Key Fields on Output (Blocked Input):

Reblock & field select — This function must be used to do those options under field select and provide unblocked output.

DASD FILES WITH KEY FIELDS (UNDEFINED)

Copy and display are the only valid functions that can be performed. The undefined-with-keys format is valid only for the DASD-to-DASD program and the DASD-to-printer program.

## Recording Job Control Statements on Output File

The following example shows how a file-to-file utility program can be used to include job control statements on an output file. The /* to be recorded on output must contain at least one additional punch in columns 3-80, since a /* card with blanks in columns 3-80 signifies end-of-file for the utilities.

```
// JOB CARDTAPE
// ASSGN SYS004,X'00C'
// ASSGN SYS005,X'282'
// UPSI 10100000
// EXEC CDTP
// UCT TC,FF,A=(80,80),B=(80,80),OU
// END
// JOB MY JCL
// EXEC ASSEMBLY
    .
    .                    data
    .
/* ABC
/&
/*                       EOF and
/&                       EOJ for utility
```

# Part 1. File—to—File Utility Programs

# Card to Disk

The card-to-disk program transfers the contents of a card file from cards to an area of disk. The cards may be punched in EBCDIC or in binary. The input records must be fixed-length unblocked, and each logical record must fit on one card. The maximum-size input record is 80 bytes, or 160 bytes for binary. These files may be copied, reblocked, field selected, or reblocked and field selected.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FF,A=(80,80),B=(80,80),I1,CY,R1,
   E=(2311)
```

The format and entries for the utility modifier statement for this program are:

```
//bUCDbTt,Ff,A=(input),B=(output),Ix,Ox,
   C=(x,y),Rx,E=(e)
```

Figure 7 shows detailed information of the entries in the utility modifier statement for the card-to-disk program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| CDb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected. For binary records this number is relative to the record as it appears in main storage, not on the card.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

When a field is to be placed into a key field (disk output), the letter K followed by a comma and the starting position of the field in the output record must be placed in parentheses.

Example: //bFSbr,s,(K,t)

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

```
r,(P,n,m),t
```

P
identifies the pack operation.

n
is the size of the input field.

m
is the size of the output field.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF | F | The initial F of this form identifies this as the format parameter. |
| | | F | The second F of the form must be indicated for fixed-length records. |
| Input Description | A=(n,m) | A= | The letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| Output Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| | B=(K=l,D=l) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (K=l,D=l) | For fixed-length disk output records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| Card Input<br>Ix | I1<br>I2 | I | The first letter in these forms identifies this parameter. |
| | | 1 | EBCDIC Input |
| | | 2 | Binary Input |
| Disk Check<br>Ox | OY<br>ON | O | The first letter in these forms identifies this parameter. |
| | | Y | Write-disk check. |
| | | N | Do not write-disk check. |
| Sequence-Numbering<br>Q=(x,y) | Q=(x,y) | Q= | This letter and symbol identify this parameter. |
| | | x | This represents the first position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits). |
| | | , | Separator |
| | | y | This represents the length of the field (maximum 10). The (x,y) portion of this parameter must be enclosed in parentheses. |
| First Record<br>Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be by-passed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Device Type Description | E=(e) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (e) | For output devices the valid entries are 2311 and 2314. This entry must be enclosed in parentheses. If a 2319 is used, 2314 must be specified. |

Figure 7. Card-to-Disk Utility Modifier Statement

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
: identifies the unpack operation.

n
: is the size of the input field.

m
: is the size of the output field.

## Control Statement Stream

A sample control statement input stream for running the card-to-disk program from the relocatable disk resident library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYSLNK,X'190'
//bASSGNbSYS001,X'180'
//bASSGNbSYS004,X'00A'
//bASSGNbSYS009,X'191'
//bOPTIONbLINK
bINCLUDEbIJWCD
bPHASEbCDDK5,IJWCDCS2,NOAUTO
bINCLUDEbUSERLABR (module name for user's
   label processing routine)
bENTRY
//bEXECbLNKEDT
//bUPSIb00010000 (standard and
   user-standard labels on output)
//bDLBLbUOUT,'DISKbFILE',67/300
//bEXTENTbSYS009,006801,8,0,01204,00204
//bEXEC
//bUCDbTF,FF,A=(80,80),B=(K=26,D=80)
//bFSb1,80,1/30,26,(K,1)
//bEND
   (data cards on SYS004)
/*
/&
```

# Card to Printer and/or Punch

Input records to this program must be fixed-length and unblocked. Card input and output can be either EBCDIC or binary, except for both printing and punching when it must be EBCDIC. The maximum record size is 80 bytes for EBCDIC, 160 bytes for binary. Card to punch requires the 1/2-4 burst mode switch of the 2821 to be in 2-4, or burst setting, to allow maximum throughput speed on the 2540.

## Card to Printer

The card-to-printer program can produce printed output in two formats: display and list. Sequence checking is performed on the input.

DISPLAY

The card-to-printer program with the display option transfers the contents of a card file to a printer with each record being placed on one print line. The field select option cannot be performed with display. In this format the first 20 positions of the print line are reserved for information describing the file. When hexadecimal printout is called for, the entire card is printed on two lines.

LIST

The input records to this program are transferred to the printer with each record being fully printed. The field select option may be used. The full print line is available for printing. When hexadecimal printout is called for, the output record size is bound by the size of the print line.

## Card to Punch

In the card-to-punch program records may be copied or field selected. Sequence fields are generated but input is not checked.

## Card to Printer and Punch

This program produces printed output in the list format. Sequence fields are generated but input is not checked.

## Utility Modifier Statement

This statement contains information required for the operation of the program. If this statement is omitted from the program, both printing and punching are performed (a printer and a punch must be assigned). The following parameters are then assumed with respect to the particular output:

Card to Punch:

```
//bUbTC,FF,A=(80,80),B=(80,80),I1,O1,S2,R1
```

Card to Printer:

```
//bUbTL,FF,A=(80,80),B=(120),I1,OC,PY,S2,R1
```

The format and entries for the utility modifier statement are:

Card to Punch:

```
//bUCPbTt,Ff,A=(n,m),B=(n,m),Ix,Ox,Rx,Sx,
   Q=(x,y)
```

Card to Printer:

```
//bUCPbTt,Ff,A=(n,m),B=(p),Ix,Ox,Px,Rx,Sx,
   Q=(x,y)
```

Card to Printer and Punch:

```
//bUCPbTt,Ff,A=(n,m),B=(n,m),Ix,Px,Rx,Sx,
   Q=(x,y)
```

Figure 8 shows detailed information of the entries in the utility modifier statement for the card-to-printer and/or punch program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| CPb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete in one statement.

The format and contents of this statement are: //bFSbr,s,t/r,s,t/r,s,t

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected. For binary records, this number is relative to the record as it appears in main storage, not on the card.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

r,(P,n,m),t

P
identifies the pack operation.

n
is the size of the input field.

m
is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
identifies the unpack operation.

n
is the size of the input field.

m
is the size of the output field.

Note: Field selections when running card to printer and/or punch will be reflected on the output for both.

HEXADECIMAL

When a program has printed output, the field selected may be printed in hexadecimal representation. This operation is indicated as follows:

r,(X,n),t

X
identifies the hexadecimal operation;

n
is the size of the input field.

Only the field length of the input is necessary for this operation because the output length will always be assumed to be twice as large. X and n are enclosed in parentheses and separated by a comma.

## Control Statement Stream

A sample control statement input stream for running the card-to-printer and/or punch program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'00C'      (reader)
//bASSGNbSYS005,X'00E'      (printer)
//bASSGNbSYS006,X'00D'      (punch)
//bEXECbCDPP
//bEND                      (assumed values)
   (data cards on SYS004)
/*
/&
```

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TB<br>TBF<br>TC<br>TD<br>TF<br>TL<br>TLF | T | The initial T identifies this as the type of function parameter. |
| | | B | Both Print and Punch |
| | | BF | Both Print and Punch with Field Select |
| | | C | Copy (punch output only) |
| | | D | Display |
| | | F | Field Select (punch output only) |
| | | L | List |
| | | LF | List and Field Select |
| Format<br>Ff | FF | F | The initial F of this form identifies this as the format parameter. |
| | | F | The second F of the form must be indicated for fixed-length records. |
| Input<br>Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| Output<br>Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| | B=(p) | B= | This letter and symbol indicate this as the output-description parameter. |
| | | (p) | For printer output, the size of the print line (120, 132, or 144) must be entered. A print line size of 120 is forced if TD is specified. |
| Card Input<br>Ix | I1<br>I2 | I | The first letter in these forms identifies this parameter. |
| | | 1 | EBCDIC Input |
| | | 2 | Binary Input |
| Printer or<br>Punch Output<br>Ox | O1<br>O2<br>OX<br>OC | O | The first letter in these forms identifies this parameter. For printer output, the type of output indicated by the field-select parameter (hexadecimal or character) overrides this parameter. |
| | | 1 | EBCDIC Output (punch only) |
| | | 2 | Binary Output (punch only) |
| | | X | Hexadecimal Output (printer only) |
| | | C | Character Output (printer only) |
| Page<br>Numbering<br>Px | PY<br>PN | P | The first letter in these forms identifies this parameter. |
| | | Y | Number pages. |
| | | N | Do not number pages. |

Figure 8. Card-to-Printer and/or Punch Utility Modifier Statement (Part 1 of 2)

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Sequence-Numbering Q=(x,y) | Q=(x,y) | Q= | This letter and symbol identify this parameter. |
| | | x | This represents the first position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits). |
| | | , | Separator |
| | | y | This represents the length of the field (maximum 10). The (x,y) portion of this parameter must be enclosed in parentheses. |
| First Record Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be by-passed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Spacing and Stacker Control Sx | S1<br>S2<br>S3 | S | The first letter in these forms identifies this parameter. |
| | | 1 | Printer Output: Single Spacing<br>Punch Output: Select Stacker 1<br>Printer and Punch: Printer Control Only |
| | | 2 | Printer Output: Double Spacing<br>Punch Output: Select Stacker 2<br>Printer and Punch: Printer Control Only |
| | | 3 | Printer Output: Triple Spacing<br>Punch Output: Invalid<br>Printer and Punch: Printer Control Only |

Figure 8. Card-to-Printer and/or Punch Utility Modifier Statement (Part 2 of 2)

# Card to Tape

The card-to-tape program transfers the contents of a card file from cards to tape. The cards may be punched in EBCDIC or binary. The input records must be fixed-length unblocked, and each logical record must fit on one card. The maximum size record is 80 bytes, or 160 for binary. These files may be copied, reblocked, field selected, or reblocked and field selected.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this card is omitted from the program, the following parameters are assumed:

```
//bUbTC,FF,A=(80,80),B=(80,80),I1,CU,R1
```

The format and entries for the utility modifier statement for this program are:

```
//bUCTbTt,Ff,A=(input),B=(output),
   Ix,Rx,Ox,Q=(x,y)
```

Figure 9 shows detailed information of the entries in the utility modifier statement for the card-to-tape program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as the utility modifier statement. |
| CTb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete on one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected. For binary records, this number is relative to the record as it appears in main storage, not on the card.<br><br>, (comma) separates entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one of the output record.<br><br>/ (slash) separates selected fields. |

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

```
r,(P,n,m),t
```

p
: identifies the pack operation.

n
: is the size of the input field.

m
: is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

```
r,(U,n,m),t
```

U
: identifies the unpack operation.

n
: is the size of the input field.

m
: is the size of the output field.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T<br><br>C<br><br>F<br><br>R<br><br>RF | The initial T identifies this as the type of function parameter.<br><br>Copy<br><br>Field Select<br><br>Reblock<br><br>Reblock and Field Select |
| Format<br>Ff | FF | F<br><br>F | The leading F of this form identifies this as the format parameter.<br><br>The second F of the form must be indicated for fixed-length records. |
| Input<br>Description | A=(n,m) | A=<br><br>(n,m) | This letter and symbol indicate this is the input-description parameter.<br><br>For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| Output<br>Description | B=(n,m) | B=<br><br>(n,m) | This letter and symbol indicate this is the output-description parameter.<br><br>For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| Card Input<br>Ix | I1<br>I2 | I<br><br>1<br><br>2 | The first letter in these forms identifies this parameter.<br><br>EBCDIC Input<br><br>Binary Input |
| Rewind<br>Output<br>Ox | OR<br>ON<br>OU | O<br><br>R<br><br>N<br><br>U | The first letter in these forms identifies this parameter. The rewind option for the output tape is active both before and after data transfer.<br><br>Rewind both before and after data transfer.<br><br>Do not rewind either before or after data transfer.<br><br>Rewind before and rewind and unload after data transfer. |
| Sequence-<br>Numbering<br>Q=(x,y) | Q=(x,y) | Q=<br><br>x<br><br>,<br><br>y | This letter and symbol identify this parameter.<br><br>This represents the first position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits).<br><br>Separator.<br><br>This represents the length of the field (maximum 10). The (x,y) portion of this parameter must be included in parentheses. |
| First Record<br>Rx | Rx | R<br><br>x | The first letter in this form identifies this parameter.<br><br>This represents the position of the first logical input record to be output (x-1 records will be by-passed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |

Figure 9. Card-to-Tape Utility Modifier Statement

## Control Statement Stream

A sample control statement input stream for running a card-to-tape program from the disk resident relocatable library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYSLNK,X'190'
//bASSGNbSYS001,X'180'
//bOPTIONbLINK
bINCLUDEbIJWCT
bPHASEbCDTP5,IJWCTCS2,NOAUTO
bINCLUDEbIJWLAB
bENTRY
//bLBLTYPbTAPE
//bEXECbLNKEDT
//bUPSIb00101000 (unlabeled output with
   no tapemark at start of file)
//bASSGNbSYS004,X'00C'  (reader)
//bASSGNbSYS005,X'183',ALT (alternate tape
                                   drive)
//bEXEC
//bUCTbTR,FF,A=(80,80),B=(80,800),CR
//bEND
   (data goes in SYS004)
/*
/&
```

A sample control statement input stream for running a card-to-tape program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'00C' (reader)
//bASSGNbSYS005,X'182' (tape drive)
   (assume standard labels)
//bEXECbCDTP
//bEND
   (data goes in SYS004)
/*
/&
```

# Data Cell to Data Cell

The data cell-to-data cell program transfers files between any number of assigned data cell units or between areas of the same unit. Using the same device for input and output can cause a reduction in performance. Files can be copied, reblocked, field selected, or reblocked and field selected. If the field select or reblock options are used, the input records must be fixed length or variable length.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FU,A=(1000),B=(1000),OY,R1,
   E=(2321,2321)
```

The format and entries for the utility modifier statement for this program are:

```
//bUMMbTt,Ff,A=(input),B=(output),Cx,Rx,
   E=(c,d)
```

Figure 10 shows detailed information of the entries in the utility modifier statement for the data cell-to-data cell program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| MMb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select statements follow. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

When a field is to be selected from a key field (data cell input), the letter K followed by a comma and the starting position of the field to be selected must be placed in parentheses.

Example: `//bFSb(K,r),s,t`

When a field is to be selected from a key field (data cell input) and is to be placed into a key field (data cell output), the starting position of the field in the input record and output record must be preceded by the letter K and a comma and enclosed in parentheses.

Example: `//bFSb(K,r),s,(K,t)`

When a field is to be placed into a key field (data cell output), the letter K followed by a comma and the starting position of the field in the output record must be placed in parentheses.

Example: `//bFSbr,s,(K,t)`

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable-length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| | A=(K=I,D=I) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=I,D=I) | For fixed-length data cell input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| Output Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable-length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | B=(K=I,D=I) | B= | This letter and symbol indicate this is the output description parameter. |
| | | (K=I,D=I) | For fixed-length data cell output records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (g) | For undefined output records or variable output records without field select, the maximum block length must be enclosed in parentheses. |
| Data Cell Check<br>Ox | OY<br>ON | O | The first letter in these forms identifies this parameter. |
| | | Y | Write-data cell check (forced for this program). |
| | | N | Do not write-data cell check (ignored for this program). |
| First Record<br>Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be by-passed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Device Type Description | E=(c,d) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (c,d) | For input devices (the letter c) and output devices (the letter d) the valid entry is 2321. These entries must be enclosed in parentheses and separated by a comma. |
| | E=(e) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (e) | For input and output devices the valid entry is 2321. This entry must be enclosed in parentheses. |

Figure 10. Data Cell-to-Data Cell Utility Modifier Statement

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

r,(P,n,m),t

P
identifies the pack operation.

n
is the size of the input field.

m
is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
identifies the unpack operation.

n
is the size of the input field.

m
is the size of the output field.

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running a data cell-to-data cell program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS010,X'193'
//bUPSIb00000000  (standard labels)
//bDLBLbUIN,'DCbFILE',67/365
//bEXTENTbB=4,SYS010,010203,1,0,19006,00094
//bDLBLbUCUT,'DATAbCELLbCUTPUT',67/300
//bEXTENTbB=5,SYS010,000123,1,0,19006,00094
//bEXECbDCDC
//bUMMbTF,FF,A=(80,80),B=(K=10,D=70),ON
//bFSb75,6,(K,1)/1,70,1
//bEND
/&
```

Note: If you want to process sequential files to support forced-end-of-volume (FEOV), bit 7 of the UPSI byte must be set on (//bUPSIb00000001). This bit 7 must be set off, if the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).

# Data Cell to Disk

The data cell-to-disk program transfers a file between any number of assigned data cells and disks. Files can be copied, reblocked, field selected, or reblocked and field selected. If the field select or reblock options are to be used, the input records must be fixed or variable length.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FU,A=(1000),B=(1000),OY,R1,
   E=(2321,2311)
```

The format and entries for the utility modifier statement for this program are:

```
//bUMDbTt,Ff,A=(input),B=(output),Cx,Rx,
   E=(c,d)
```

Figure 11 shows detailed information of the entries in the utility modifier statement for the data cell-to-disk program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| MDb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete on one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

When a field is to be selected from a key field (data cell input), the letter K followed by a comma and the starting position of the field to be selected must be placed in parentheses.

Example: //bFSb(K,r),s,t

When a field is to be selected from a key field (data cell input) and is to be placed into a key field (disk output), the starting position of the field in the input record and output record must be preceded by the letter K and a comma and enclosed in parentheses.

Example: //bFSb(K,r),s,(K,t)

When a field is to be placed into a key field (disk output), the letter K followed by a comma and the starting position of the field in the output record must be placed in parentheses.

Example: //bFSbr,s,(K,t)

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F of the first possible form must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| | A=(K=I,D=I) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=I,D=I) | For fixed-length data cell input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| Output Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | B=(K=I,D=I) | B= | This letter and symbol indicate this is the output description parameter. |
| | | (K=I,D=I) | For fixed-length disk output records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (g) | For undefined output records or variable output records without field select, the maximum block length must be enclosed in parentheses. |
| Disk Check<br>Ox | OY<br>ON | O | The first letter in these forms identifies this parameter. |
| | | Y | Write-disk check. |
| | | N | Do not write-disk check. |
| First Record<br>Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be bypassed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Device Type Description | E=(c,d) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (c,d) | For input devices (the letter c) the valid entry is 2321. For output devices (the letter d) the valid entries are 2311 and 2314. If a 2319 is used, 2314 must be specified.<br>These entries must be enclosed in parentheses and separated by a comma. |

Figure 11. Data Cell-to-Disk Utility Modifier Statement

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

r,(P,n,m),t

P
identifies the pack operation.

n
is the size of the input field.

m
is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
identifies the unpack operation.

n
is the size of the input field.

m
is the size of the output field.

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running a data cell-to-disk program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS010,X'193'
//bASSGNbSYS015,X'191'
//bUPSIb00000000  (standard labels)
//bDLBLbUIN,'EXAMPLEbFILE',67/365
//bEXTENTbB=4,SYS010,010203,1,0,12040,00060
//bDLBLbUOUT,'DISKbFILEbEXAMPLE',67/300
//bEXTENTbSYS010,000123,1,0,01502,00038
//bEXECbDCDK
//bUMDbTC,FF,A=(K=10,D=100),B=(K=10,D=100),
   OY
//bEND
/&
```

Note: If you want to process sequential files to support forced-end-of-volume (FEOV), bit 7 of the UPSI byte must be set on (//bUPSIb00000001). This bit 7 must be set off, if the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).

# Data Cell to Printer

The data cell-to-printer program can display a data cell file in either data display or data list format. Data display provides a visual picture of the data where every byte appears in the printed output; this format can handle fixed, variable, and undefined records. Data list provides a simple edited list of the file. The input file can come from one or more data cells. If data list is used, input records must be fixed or variable length and (for the data list function only) must not exceed the size of the print line.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTD,FU,A=(1000),B=(120),CX,S1,PY,R1,
   E=(2321)
```

The format and entries for the utility modifier statement for this program are:

```
//bUMPbTt,Ff,A=(input),B=(output),Cx,Sx,Px,
   Rx,E=(e)
```

Figure 12 shows detailed information of the entries in the utility modifier statement for the data cell-to-printer program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this card as a utility modifier statement. |
| MPb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. This is valid only for data list mode. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the print line.<br><br>/ (slash) separates selected fields. |

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

```
r,(U,n,m),t
```

U
: identifies the unpack operation.

n
: is the size of the input field.

m
: is the size of the output field.

HEXADECIMAL

When a program has printed output, the field selected may be printed in hexadecimal representation. This operation is indicated as follows:

```
r,(X,n),t
```

X
: identifies the hexadecimal operation.

n
: is the size of the input field. Only the field length of the input is necessary for this operation because the output length will always be assumed to be twice as large. X and n are enclosed in parentheses and separated by a comma.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TD<br>TL<br>TLF | T | The initial T identifies this as the type of function parameter. |
| | | D | Display |
| | | L | List |
| | | LF | List and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records.<br>With undefined records, the only valid entry above is data display. |
| Input<br>Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(K=I,D=I) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=I,D=I) | For fixed-length data cell input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| Output<br>Description | B=(p) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (p) | For printer output, the size of the print line (120, 132, or 144) must be entered.<br>A print line size of 120 is forced, if TD is specified. |
| | B=(n,p) | B= | This letter and symbol indicate this is the output description parameter. |
| | | (n,p) | For field select of variable length records with printer output records the fixed portion of each output record (the letter n) and the size of the print line (the letter p) must be enclosed in parentheses and separated by a comma. |
| Printer<br>Output<br>Ox | OX<br>OC | O | The first letter in these forms identifies this parameter. The character printout is forced for data list. The type of output indicated by the field-select parameter (hexadecimal or character) overrides this parameter. |
| | | X | Hexadecimal Printout |
| | | C | Character Printout |
| Page-<br>Numbering<br>Px | PY<br>PN | P | The first letter in these forms identifies this parameter. |
| | | Y | Number pages (forced for data display). |
| | | N | Do not number pages (forced for first character forms control). |
| First Record<br>Printed<br>Rx | Rx | R | The first letter in these forms identifies this parameter. |
| | | x | This represents the position of the first logical record to be printed; x-1 records will be bypassed. |

Figure 12. Data Cell-to-Printer Utility Modifier Statement (Part 1 of 2)

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Spacing<br>Sx | S1<br>S2<br>S3<br>SA<br>SB<br>SC<br>SD | S<br>1<br>2<br>3<br>A<br>B<br>C<br>D | The first letter in these forms identifies this parameter.<br>Single Spacing (forced for data display)<br>Double Spacing<br>Triple Spacing<br>Type A First Character Forms Control<br>Type B First Character Forms Control<br>Type C First Character Forms Control<br>Type D First Character Forms Control |
| Device Type Description | E=(e) | E=<br>(e) | This letter and symbol indicate this is the device type description parameter.<br>For input devices the valid entry is 2321. This entry must be enclosed in parentheses. |

Figure 12. Data Cell-to-Printer Utility Modifier Statement (Part 2 of 2)

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running a data cell-to-printer program from the core image library follows; devices and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS010,X'193'
//bASSGNbSYS005,X'00E'
   (assume standard labels)
//bDLBLbUIN,'EXAMPLEbFILE',67/365
//bEXTENTbB=3,SYS010,001101,1,0,12100,00900
//bEXTENTbB=3,SYS010,001101,1,1,16000,00013
//bEXECbDCPR
//bUMPbTL,FF,A=(K=20,D=90),B=(120),OC,S2,PN
//bH1bLISTbOFbDATAbCELLbFILE
//bEND
/&
```

Note: If you want to process sequential files to support forced-end-of-volume (FEOV), bit 7 of the UPSI byte must be set on (//bUPSIb00000001). This bit 7 must be set off, if the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).

# Data Cell to Tape

The data cell-to-tape program transfers a file from one or more data cells to one or more tape reels. These files may be copied, reblocked, field selected, or reblocked and field selected. If either the field select or reblock option is to be used, the input records must be fixed or variable-length.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FU,A=(1000),B=(1000),OU,R1,E=(2321)
```

The format and entries for the utility modifier statement for this program are:

```
//bUMTbTt,Ff,A=(input),B=(output),Cx,Rx,
   E=(e)
```

Figure 13 shows detailed information of the entries in the utility modifier statement for the data cell-to-tape program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| MTb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field-select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

When a field is to be selected from a key field (data cell input), the letter K followed by a comma and the starting position of the field to be selected must be placed in parentheses.

Example: //bFSb(K,r),s,t

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

```
r,(P,n,m),t
```

P
: identifies the pack operation.

n
: is the size of the input field.

m
: is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

```
r,(U,n,m),t
```

U
: identifies the unpack operation.

n
: is the size of the input field.

m
: is the size of the output field.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F of the first possible form must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input<br>Description | A=(n,m) | A= | This letter and symbol indicate this is the input description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(K=I,D=I) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=I,D=I) | For fixed-length data cell input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| Output<br>Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (g) | For undefined output records or variable output records without field select, the maximum block length must be enclosed in parentheses. |
| Rewind Output | OR<br>ON<br>OU | O | The first letter in these forms identifies this parameter.<br>The rewind option for the output tape is active both before and after data transfer. |
| | | R | Rewind both before and after data transfer. |
| | | N | Do not rewind either before or after data transfer. |
| | | U | Rewind before and rewind and unload after data transfer. |
| First Record<br>Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be bypassed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Device Type<br>Description | E=(e) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (e) | For input devices the valid entry is 2321. This entry must be enclosed in parentheses. |

Figure 13. Data Cell-to-Tape Utility Modifier Statement

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running a data cell-to-tape program from the disk resident relocatable library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS010,X'193'
//bASSGNbSYS005,X'182'
//bASSGNbSYSLNK,X'190'
//bASSGNbSYS001,X'180'
//bUPSIb01000000 (user standard input and
                  standard output labels)
//bOPTIONbLINK
bINCLUDEbIJWMT
bPHASEbDCTP5,IJWMTCS2,NOAUTO
bINCLUDEbULABROUT (user label processing
                   routine)
bENTRY
//bLBLTYPbTAPE
//bEXECbLNKEDT
//bTLBLbUCUT,'EXAMPLEbFILE',67/300,010203,
   0001,0001,0001,01
//bDLBLbUIN,'DATAbCELLbTObTAPE',67/365
//bEXTENTbB=9,SYS010,000123,1,0,12680,00020
//bEXEC
//bUMTbTRF,FF,A=(K=10,D=100),B=(110,440),ON
//bFSb(K,1),10,1/1,100,11
//bEND
/&
```

Note: If you want to process sequential files to support forced-end-of-volume (FEOV), bit 7 of the UPSI byte must be set on (//bUPSIb00000001). This bit 7 must be set off, if the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).

# Disk to Card

The disk-to-card program transfers the contents of a disk file to a card file. The output file may be punched in either EBCDIC or binary. Each logical output record must fit on one card (80 bytes for EBCDIC or 160 bytes for binary). Unless only a portion of the input record is transferred through the field select option, the input record size will be restricted to 80 or 160. Input records to this program must be fixed length. Files in this program may be copied, reblocked, field selected, or reblocked and field selected. Blocked input records must be reblocked.

## Sequence Numbering

Sequence numbering of the cutput tc this program may be requested. A field up to ten characters in length will be punched into each card. This field will be numbered starting from one (with high-order zeros), and will be increased by one fcr each succeeding card. In the event that a sufficiently long field is nct defined tc number all of the cards, the numbers will wrap around to zero with nc error indication. This option is independent of field select. The sequence number will overlay any data selected into the sequence area of the card.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FF,A=(80,80),B=(80,80),O1,R1,S2,
   E=(2311)
```

The format and entries for the utility modifier statement for this program are:

```
//bUDCbTt,Ff,A=(input),B=(cutput),Cx,Rx,
   Sx,Q=(x,y),E=(e)
```

Figure 14 shows detailed information of the entries in the utility modifier statement for the disk-to-card program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as the utility modifier statement. |
| DCb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-tc-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follcw. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected. For binary records, this number is relative to the record as it appears in main storage, not on the card.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF | F | The leading F of this form identifies this as the format parameter. |
| | | F | The second F of the form must be indicated for fixed-length records. |
| Input<br>Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| | A=(K=l,D=l) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=l,D=l) | For fixed-length disk input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in a parentheses. |
| Output<br>Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| Output Mode<br>Ox | O1<br>O2 | O | The first letter in these forms identifies this parameter. |
| | | 1 | EBCDIC Punching |
| | | 2 | Binary Punching |
| Sequence-<br>Numbering<br>Q=(x,y) | Q=(x,y) | Q= | This letter and symbol identify this parameter. |
| | | x | This represents the first position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits). |
| | | , | Separator |
| | | y | This represents the length of the field (maximum 10). The (x,y) portion of this parameter must be enclosed in parentheses. |
| First Record<br>Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be by-passed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Stacker<br>Select<br>Sx | S1 | S | The first letter in these forms identifies this parameter. |
| | S2 | 1 | Select Pocket 1 |
| | S3 | 2 | Select Pocket 2 |
| | | 3 | First Character Stacker Select |
| Device Type<br>Description | E=(e) | E= | This letter and sy mbol indicate this is the device type description parameter. |
| | | (e) | For input devices the valid entries are 2311 and 2314. If a 2319 is used, 2314 must be specified. This entry must be enclosed in parentheses. |

Figure 14. Disk-to-Card Utility Modifier Statement

When a field is to be selected from a key field (disk input), the letter K followed by a comma and the starting position of the field to be selected must be placed in parentheses.

Example: //bFSb(K,r),s,t

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

r, (P,n,m),t

P
: identifies the pack operation.

n
: is the size of the input field.

m
: is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
: identifies the unpack operation.

n
: is the size of the input field.

m
: is the size of the output field.

## Control Statement Stream

A sample control statement input stream for running a disk-to-card program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS006,X'00D'
//bASSGNbSYS009,X'191' (standard labels
                        assumed)
//bDLBLbUIN,'DISKbFILE',67/365
//bEXTENTbSYS009,112233,1,0,01400,00043
//bEXECbDKCD
//bEND
/&
```

Note: If you want to process sequential files to support forced-end-of-volume (FEOV), bit 7 of the UPSI byte must be set on (//bUPSIb00000001). This bit 7 must be set off, if the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).

# Disk to Data Cell

The disk-to-data cell program transfers a file between any number of assigned data cells and disks. Files can be copied, reblocked, field selected, or reblocked and field selected. If the field select or reblock options are to be used, the input records must be fixed length or variable length.

This program may also be used to condense a sequential disk file earlier created with a forced end-of-volume record. A FEOV record will therefore not appear on the output file. For a full description of forced end of volume, see the DOS Supervisor and I/O Macros publication.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FU,A=(1000),B=(1000),OY,R1,
   E=(2311,2321)
```

The format and entries for the utility modifier statement for this program are:

```
//bUDMbTt,Ff,A=(input),B=(output),Cx,Rx,
   E=(c,d)
```

Figure 15 shows detailed information of the entries in the utility modifier statement for the disk-to-data cell program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| DMb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If cards are punched, each card need not be filled even if additional field select statements follow. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

When a field is to be selected from a key field (disk input), the letter K followed by a comma and the starting position of the field to be selected must be placed in parentheses.

Example: //bFSb(K,r),s,t

When a field is to be selected from a key field (disk input) and is to be placed into a key field (data cell output), the starting position of the field in the input record and output record must be preceded by the letter K and a comma and enclosed in parentheses.

Example: //bFSb(K,r),s,(K,t)

When a field is to be placed into a key field (data cell output), the letter K followed by a comma and the starting position of the field in the output record must be placed in parentheses.

Example: //bFSbr,s,(K,t)

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

r,(P,n,m),t

P
identifies the pack operation.

n
is the size of the input field.

m
is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
identifies the unpack operation;

n
is the size of the input field;

m
is the size of the output field.

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running the disk-to-data cell program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bPAUSEbMOUNTbPACKb352336bONb190bANDbCELLb
   362431bONbBINb4bOFb193
//bASSGNbSYS004,X'190'
//bASSGNbSYS005,X'193' (standard labels
                        assumed)
//bDLBLbUIN,'DISKbFILE',67/365
//bEXTENTbSYS004,352336,1,0,00551,00082
//bDLBLbUOUT,'BACK-UPbFILE',67/300
//bEXTENTbB=4,SYS005,362437,1,0,01465,00032
//bEXECbDKDC
//bUDMbTC,FU,A=(960),B=(960),OY
//bEND
/&
```

Note: If you want to process sequential files to support forced-end-of-volume (FEOV), bit 7 of the UPSI byte must be set on (//bUPSIb00000001). This bit 7 must be set off, if the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F of the first possible form must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| | A=(K=l,D=l) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=l,D=l) | For fixed-length disk input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| Output Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | B=(K=l,D=l) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (K=l,D=l) | For fixed-length data cell output records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (g) | For undefined output records or variable output records without field select, the maximum block length must be enclosed in parentheses. |
| Data Cell Check Ox | OY<br>ON | O | The first letter in these forms identifies this parameter. |
| | | Y | Write-data cell check (forced for this program). |
| | | N | Do not write-data cell check (ignored for this program). |
| First Record Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be by-passed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Device Type Description | E=(c,d) | E=<br>(c,d) | This letter and symbol indicate this is the device type description parameter.<br>For input devices (the letter c) the valid entries are 2311 and 2314. If a 2319 is used, 2314 must be specified. For output devices (the letter d) the valid entry is 2321. These entries must be enclosed in parentheses and separated by a comma. |

Figure 15. Disk-to-Data Cell Utility Modifier Statement

# Disk to Disk

The disk-to-disk program transfers a file between disk units or between areas of the same unit. Using the same device for input and output can cause a reduction in performance. Files can be copied, reblocked, field selected, or reblocked and field selected. If the field select or reblock options are to be used, the input records must be fixed or variable length.

This program may also be used tc condense a sequential disk file earlier created with a forced end-of-volume record. A FEOV record will therefore not appear on the output file. For a full description of forced end of volume, see the DOS Supervisor and I/O Macros publication.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FU,A=(1000),B=(1000),OY,R1,
   E=(2311,2311)
```

The format and entries for the utility modifier statement for this program are:

```
//bUDDbTt,Ff,A=(input),B=(cutput),Cx,Rx,
   E=(c,d)
```

Figure 16 shows detailed information of the entries in the utility modifier statement for the disk-to-disk program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| DDb | The initials of the program. These initials can be omitted if the statement is to be used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need nct be filled even if additional field select cards follow. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

When a field is to be selected from a key field (disk input), the letter K followed by a comma and the starting position of the field to be selected must be placed in parentheses.

Example: //bFSb(K,r),s,t

When a field is to be selected from a key field (disk input) and is to be placed into a key field (disk output), the starting position of the field in the input record and output record must be preceded by the letter K and a comma and enclosed in parentheses.

Example: //bFSb(K,r),s,(K,t)

When a field is to be placed into a key field (disk output), the letter K followed by a comma and the starting position of the field in the output record must be placed in parentheses.

Example: //bFSbr,s,(K,t)

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F of the first possible form must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input<br>Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| | A=(K=I,D=I) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=I,D=I) | For fixed-length disk input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| Output<br>Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | B=(K=I,D=I) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (K=I,D=I) | For fixed-length disk output records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (g) | For undefined output records or variable output records without field select, the maximum block length must be enclosed in parentheses. |
| Disk Check<br>Ox | OY<br>ON | O | The first letter in these forms identifies this parameter. |
| | | Y | Write-disk check. |
| | | N | Do not write-disk check. |
| First Record<br>Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be bypassed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Device Type<br>Description | E=(c, d) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (c, d) | For input devices (the letter c) and output devices (the letter d) the valid entries are 2311 and 2314. If a 2319 is used, 2314 must be specified. These entries must be enclosed in parentheses and separated by a comma. |
| | E=(e) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (e) | For input and output devices the valid entries are 2311 or 2314. This entry must be enclosed in parentheses. |

Figure 16. Disk-to-Disk Utility Modifier Statement

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

r,(P,n,m),t

P
identifies the pack operation.

n
is the size of the input field;

m
is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
identifies the unpack operation.

n
is the size of the input field.

m
is the size of the output field.

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running a disk-to-disk program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bPAUSEbMOUNTbPACKbONbDRIVEb191
//bASSGNbSYS004,X'191'
//bASSGNbSYS005,X'191' (standard labels
                        assumed)
//bDLBLbUIN,'DISKbINPUT',67/365
//bEXTENTbSYS004,222333,1,0,00913,00020
//bDLBLbUOUT,'DISKbOUTPUT',67/300
//bEXTENTbSYS005,222333,1,0,00413,00020
//bEXECbDKDK
//bUDDbTR,FF,A=(80,80),B=(80,960),OY
//bEND
/&
```

Note: If you want to process sequential files to support forced-end-of-volume (FEOV), bit 7 of the UPSI byte must be set on (//bUPSIb00000001). This bit 7 must be set off, if the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).

# Disk to Printer

The disk-to-printer program can display a disk file in either data display or data list format. Data display provides a visual picture of the data where every byte appears in the printed output; this format can handle fixed, variable, and undefined records. Data list provides a simple edited list of the file; input records must be fixed or variable length and (for the data list function only) must not exceed the size of the print line. An option is available to this program to specify the number of logical records in a file to be bypassed before printing begins.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTD,FU,A=(1000),B=(120),OX,S1,PY,R1,
   E=(2311)
```

The format and entries for the utility modifier statement for this program are:

```
//bUDPbTt,Ff,A=(input),B=(output),Ox,Sx,
   Px,Rx,E=(e)
```

Figure 17 shows detailed information of the entries in the utility modifier statement for the disk-to-printer program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| DPb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. This is valid only for data list mode. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the print line.<br><br>/ (slash) separates selected fields. |

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

```
r,(U,n,m),t
```

U
: identifies the unpack operation.

n
: is the size of the input field.

m
: is the size of the output field.

HEXADECIMAL

When a program has printed output, the field selected may be printed in hexadecimal representation. This operation is indicated as follows:

```
r,(X,n),t
```

X
: identifies the hexadecimal operation;

n
: is the size of the input field. Only the field length of the input is necessary for this operation because the output length will always be assumed to be twice as large. X and n are enclosed in parentheses and separated by a comma.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TD<br>TL<br>TLF | T | The initial T identifies this as the type of function parameter. |
| | | D | Display |
| | | L | List |
| | | LF | List and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records.<br>With undefined records, the only valid entry above is data display. |
| Input<br>Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(K=1,D=1) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=1,D=1) | For fixed-length disk input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, length must be enclosed in parentheses. |
| Output<br>Description | B=(p) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (p) | For printer output, the size of the print line (120, 132, or 144) must be entered. A print line size of 120 is forced if TD is specified. |
| | B=(n,p) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,p) | For field select of variable length records with printer output records, the fixed portion of each output record (the letter n) and the size of the print line (the letter p) must be enclosed in parentheses and separated by a comma. |
| Printer<br>Output<br>Ox | OX<br>OC | O | The first letter in these forms identifies this parameter. The type of output indicated by the field-select parameter (hexadecimal or character) overrides this parameter. |
| | | X | Hexadecimal Printout (for data display only) |
| | | C | Alphameric Printout (forced for data list mode) |
| Page-<br>Numbering<br>Px | PY<br>PN | P | The first letter in these forms identifies this parameter. |
| | | Y | Number pages (forced for data display) |
| | | N | Do not number pages (forced for first character forms control) |
| First Record<br>Rx | Rx | R | The first letter in these forms identifies this parameter. |
| | | x | This represents the position of the first logical record to be printed; x - 1 records will be bypassed. |

Figure 17. Disk-to-Printer Utility Modifier Statement (Part 1 of 2)

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Spacing Sx | S1 S2 S3 SA SB SC SD | S | The first letter in these forms identifies this parameter. |
| | | 1 | Single Spacing (forced for data display) |
| | | 2 | Double Spacing |
| | | 3 | Triple Spacing |
| | | A | Type A First Character Forms Control |
| | | B | Type B First Character Forms Control |
| | | C | Type C First Character Forms Control |
| | | D | Type D First Character Forms Control |
| Device Type Description | E=(e) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (e) | For input devices the valid entries are 2311 and 2314. If a 2319 is used, 2314 must be specified. This entry must be enclosed in parentheses. |

Figure 17. Disk-to-Printer Utility Modifier Statement (Part 2 of 2)

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:
```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

INDEXED SEQUENTIAL HANDLING

An indexed sequential file can be printed by specifying sequential in the DLBL card and specifying undefined format in the utility modifier statement. The track indexes will be printed along with the prime data area. Further indexes will not be obtained.

## Control Statement Stream

A sample control statement input stream for running a disk-to-printer program from the core image library follows; device and file descriptions are peculiar to the job being run. See page 27 for a description of the heading statement.

```
//bJOBbEXAMPLE
//bASSGNbSYS007,X'191'
//bASSGNbSYS005,X'00E'
//bDLBLbUIN,'DISKbFILE',67/365
//bEXTENTbSYS007,333333,1,0,01224,00022
//bEXECbDKPR
//bUDPbTL,FF,A=(80,400),B=(132),S1
//bEND
/&
```

To print an indexed sequential file (refer to "Indexed Sequential Handling" above) follow this sample control statement input stream. Device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS007,X'191'
//bASSGNbSYS005,X'00E'
//bDLBLbUIN,'INDEXED SEQUENTIAL',69/152,SD
//bEXTENTbSYS007,333333,1,0,01224,00022
//bEXECbDKPR
//bUDPbTD,FU,A=(80),B=(132),OX
//bEND
/&
```

Note: If you want to process sequential files to support forced-end-of-volume (FEOV), bit 7 of the UPSI byte must be set on (//bUPSIb00000001). This bit 7 must be set off, if the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).

# Disk to Tape

The disk-to-tape program transfers a file from one or more disk units to one or more tape reels. These files may be copied, reblocked, field selected, or reblocked and field selected. If the field select or reblock options are used, the input records must be fixed or variable length.

This program may also be used to condense a sequential disk file earlier created with a forced end-of-volume record. A FEOV record will therefore not appear on the output file. For a full description of forced end of volume, see the DOS Supervisor and I/O Macros publication.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FU,A=(1000),B=(1000),OU,R1,E=(2311)
```

This format and entries for the utility modifier statement for this program are:

```
//bUDTbTt,Ff,A=(input),B=(output),Ox,Rx,
   E=(e)
```

Figure 18 shows detailed information of the entries in the utility modifier statement for the disk-to-tape program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| DTb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. To transfer the key field to tape, field select is required. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete on one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

When a field is to be selected from a key field (disk input), the letter K followed by a comma and the starting position of the field to be selected must be placed in parentheses.

Example: //bFSb(K,r),s,t

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

```
r,(P,n,m),t
```

P
    identifies the pack operation.

n
    is the size of the input field.

m
    is the size of the output field.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F of the first possible form must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(K=I,D=I) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (K=I,D=I) | For fixed-length disk input records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| Output Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (g) | For undefined output records or variable output records without field select, the maximum block length must be enclosed in parentheses. |
| Rewind Output<br>Ox | OR<br>ON<br>OU | O | The first letter in these forms identifies this parameter.<br>The rewind option for the output tape is active both before and after data transfer. |
| | | R | Rewind both before and after data transfer. |
| | | N | Do not rewind either before or after data transfer. |
| | | U | Rewind before and rewind and unload after data transfer. |
| First Record<br>Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be bypassed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Device Type Description | E=(e) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (e) | For input devices the valid entries are 2311 and 2314. If a 2319 is used, 2314 must be specified. This entry must be enclosed in parentheses. |

Figure 18. Disk-to-Tape Utility Modifier Statement

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
identifies the unpack operation.

n
is the size of the input field.

m
is the size of the output field.

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field-select control statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running a disk-to-tape program from the core image library follows; devices and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'191'
//bASSGNbSYS005,X'183'
//bUPSIb00100000 (tapemark at beginning of
                  unlabeled output file)
//bDLBLbUIN,'DISKbFILE',67/365
//bEXTENTbSYS004,333333,1,0,01224,00022
//bEXECbDKTP
//bUDTbTR,FF,A=(80,800),B=(80,80),OR
//bEND
/&
```

Note: If you want to process sequential files to support forced-end-of-volume (FEOV), bit 7 of the UPSI byte must be set on (//bUPSIb00000001). This bit 7 must be set off, if the file to be copied was built by using the direct access method (DAM) or indexed sequential access method (ISAM).

# Tape to Card

The tape-to-card program transfers the contents of a tape file to a card file. The output file may be punched in either EBCDIC or binary. Each logical output record must fit in one card (80 bytes for EBCDIC or 160 bytes for binary). Unless only a portion of the input record is transferred through the field select or reblock and field select option, the input record size will be restricted to 80 or 160. Input records to this program must be fixed length. These files may be copied, reblocked, field selected, or reblocked and field selected. Blocked input records must be reblocked.

## Sequence Numbering

Sequence numbering of the output tc this program may be requested. A field up to ten characters in length is punched into each card. This field is numbered starting from one (with high-order zeros) and is increased by one for each succeeding card. In the event that a sufficiently lcng field is not defined to number all of the cards, the numbers will wrap around to zero with no error indication. The sequence number will overlay any data selected intc the sequence area of the card.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FF,A=(80,80),B=(80,80),IU,C1,R1,S2
```

The format and entries for the utility modifier statement for this program are:

```
//bUTCbTt,Ff,A=(input),B=(output),Ix,Ox,
   Rx,Sx,Q=(x,y)
```

Figure 19 shows detailed information of the entries in the utility modifier statement for the tape-to-card program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| TCb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-tc-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete on one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected. For binary records this number is relative to the record as it appears in main storage, not on the card.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

```
r,(P,n,m),t
```

P
: identifies the pack operation.

n
: is the size of the input field.

m
: is the size of the output field.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF | F | The leading F of this form identifies this as the format parameter. |
| | | F | The second F of the form must be indicated for fixed-length records. |
| input<br>Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| Output<br>Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. |
| Rewind Input<br>Ix | IR<br>IN<br>IU | I | The first letter in these forms identifies this parameter. |
| | | R | Rewind both before and after data transfer. |
| | | N | Do not rewind either before or after data transfer. |
| | | U | Rewind before and rewind and unload after data transfer. |
| Sequence-Numbering<br>Q=(x,y) | Q=(x,y) | Q= | This letter and symbol identify this parameter. |
| | | x | This represents the first position of a field in a card (relative to one) for sequence-numbering (1 or 2 digits). |
| | | , | Separator |
| | | y | This represents the length of the field (maximum 10). The (x,y) parts of this parameter must be enclosed in parentheses. Absence of this parameter indicates no sequence numbers. |
| First Record<br>Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be bypassed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Stacker Control<br>Sx | S1<br>S2<br>S3 | S | The first letter in these forms identifies this parameter. |
| | | 1 | Select Pocket 1 |
| | | 2 | Select Pocket 2 |
| | | 3 | First Character Stacker Control |
| Output Mode<br>Ox | O1<br>O2 | O<br>1<br>2 | The first letter in these forms identifies this parameter.<br>EBCDIC Punching<br>Binary Punching |

Figure 19. Tape-to-Card Utility Modifier Statement

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
     identifies the unpack operation.

n
     is the size of the input field.

m
     is the size of the output field.

## Control Statement Stream

A sample control statement input stream for running a tape-to-card program from the tape resident relocatable library follows; devices and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYSLNK,X'180'
//bASSGNbSYS001,X'181'
//bASSGNbSYS002,X'182'
//bOPTIONbLINK
bINCLUDEbIJWTC
bPHASEbTPCD5,IJWTCCS2,NOAUTO
bINCLUDE
.
.
.
(user label processing routine on SYSIPT)
.
.
.
bENTRY
/*
//bLBLTYPbTAPE(01)   (01) indicates one
                      VOL-TPLAB set
//bEXECbLNKEDT
//bASSGNbSYS004,X'183'
//bASSGNbSYS006,X'00D'
//bUPSIb01000000 (standard and user
                  standard labels)
//bTLBLbUIN,'DATAbFILEb638',67/365,000121,
   0001,0001,0001,01
//bEXEC
//bUTCbTRF,FF,A=(70,700),B=(80,80),IN,S1,
   O1,R380
//bFSb1,70,1/1,10,71
//bEND
/&
```

# Tape to Data Cell

The tape-to-data cell program transfers a file from one or more tape reels tc any number of assigned data cells. These files may be copied, field selected, reblocked, or reblocked and field selected. If the field select or reblock options are used, the input records must be fixed or variable length.

## Utility Modifier Statement

This card contains information required for the operation of this program. If this card is omitted from the program, the following parameters are assumed:

```
//bUbTC,FU,A=(1000),B=(1000),IU,OY,R1,
   E=(2321)
```

The format and entries for the utility modifier statement for this program are:

```
//bUTMbTt,Ff,A=(input),B=(output),Ix,Ox,Rx,
   E=(e)
```

Figure 20 shows detailed information of the entries in the utility modifier statement for the tape-to-data cell program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| TMb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file programs to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need nct be filled even if additional field select cards follow. The field selected must be complete on one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement. |
| | FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected. |
| | , (comma) separates the entries in the parameter. |
| | s indicates the length of the field in bytes. |
| | , separator. |
| | t indicates the starting position, relative to one, of the output record. |
| | / (slash) separates selected fields. |

When a field is to be placed into a key field (data cell output), the letter K followed by a comma and the starting position of the field in the output record must be placed in parentheses.

Example: //bFSbr,s,(K,t)

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

```
r,(P,n,m),t
```

P
: identifies the pack operation.

n
: is the size of the input field.

m
: is the size of the output field.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input<br>Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma.<br><br>For field select with variable length records, the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| Output<br>Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma.<br><br>For field select with variable length records, the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | B=(K=I,D=I) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (K=I,D=I) | For fixed-length data cell output records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (g) | For undefined output records or variable output records without field select, the maximum block length must be enclosed in parentheses. |
| Rewind Input<br>Ix | IR<br>IN<br>IU | I | The first letter in these forms identifies this parameter. The rewind option for the input tape is active both before and after data transfer. |
| | | R | Rewind both before and after data transfer. |
| | | N | Do not rewind either before or after data transfer. |
| | | U | Rewind before and rewind and unload after data transfer. |
| Data Cell Check<br>Ox | OY<br>ON | O | The first letter in these forms identifies this parameter. |
| | | Y | Write-data cell check (forced for this program). |
| | | N | Do not write-data cell check (ignored for this program). |

Figure 20. Tape-to-Data Cell Utility Modifier Statement (Part 1 of 2)

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| First Record Rx | Rx | R<br>x | The first letter in this form identifies this parameter.<br>This represents the position of the first logical input record to be output (x-1 records will be bypassed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Device Type Description | E=(e) | E=<br>(e) | This letter and symbol indicate this is the device type description parameter.<br>For output devices the valid entry is 2321. This entry must be enclosed in parentheses. |

Figure 20. Tape-to-Data Cell Utility Modifier Statement (Part 2 of 2)

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
identifies the unpack operation.

n
is the size of the input field.

m
is the size of the output field.

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the cutput record, the letters CV (copy variable) rust be present in the field select statement. If this entry is rade when prccessing reccrds that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Fcllowing selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as describes in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running a tape-to-data cell program from the disk resident relocatable library follows; device and file descriptions are peculiar tc the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYSLNK,X'190'
//bASSGNbSYS001,X'180'
//bASSGNbSYS004,X'182'
//bASSGNbSYS014,X'193'
//bUPSIb10000000 (output standard label
   checking with unlabeled input)
//bOPTIONbLINK
bINCLUDEbIJWTM
bPHASEbTPCD5,IJWTMCS2,NOAUTO
bINCLUDEbIJWLAB
bENTRY
//bLBLTYPbTAPE
//bEXECbLNKEDT
//bDLBLbUOUT,'DATAbCELLbFILE',67/300
//bEXTENTbB=6,SYS014,000111,1,0,01806,
   00194
//bEXEC
//bUTMbTRF,FF,A=(110,440),B=(K=10,D=100),OY
//bFSb1,10,(K,1)/11,100,1
//bEND
/&
```

The tape-to-disk program transfers a file from one or more tape reels to a maximum of n disk units where n is the number of disk units assigned. These files may be copied, field selected, reblocked, or reblocked and field selected. If the field select or reblock options are used, the input records must be fixed or variable length.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FU,A=(1000),B=(1000),IU,OY,R1,
   E=(2311)
```

The format and entries for the utility modifier statement for this program are:

```
//bUTDbTt,Ff,A=(input),B=(cutput),Ix,Ox,Rx,
   E=(e)
```

Figure 21 shows detailed information of the entries in the utility modifier statement for the tape-to-disk program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this statement as a utility modifier statement. |
| TDb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of th field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

When a field is to be placed into a key field (disk output), the letter K followed by a comma and the starting position of the field in the output record must be placed in parentheses.

Example: //bFSbr,s,(K,t)

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

```
r,(P,n,m),t
```

P
     identifies the pack operation.

n
     is the size of the input field.

m
     is the size of the output field.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TC<br>TF<br>TR<br>TRF | T | The initial T identifies this as the type of function parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma.<br><br>For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| Output Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma.<br><br>For field select with variable length records, the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | B=(K=I,D=I) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (K=I,D=I) | For fixed-length disk output records with keys, the letter K and symbol = must precede the length of the key field. The letter D and symbol = must precede the length of the data field. These two fields must be separated by a comma and enclosed in parentheses. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (g) | For undefined output records or variable output records without field select, the maximum block length must be enclosed in parentheses. |
| Rewind Input<br>Ix | IR<br>IN<br>IU | I | The first letter in these forms identifies this parameter. The rewind option for the input tape is active both before and after data transfer. |
| | | R | Rewind both before and after data transfer. |
| | | N | Do not rewind either before or after data transfer. |
| | | U | Rewind before and rewind and unload after data transfer. |

Figure 21. Tape-to-Disk Utility Mcdifier Statement (Part 1 of 2)

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Disk Check Ox | OY ON | O | The first letter in these forms identifies this parameter. |
| | | Y | Write-disk check. |
| | | N | Do not write-disk check. |
| First Record Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be bypassed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Device Type Description | E=(e) | E= | This letter and symbol indicate this is the device type description parameter. |
| | | (e) | For output devices the valid entries are 2311 and 2314. If a 2319 is used, 2314 must be specified. This entry must be enclosed in parentheses. |

Figure 21. Tape-to-Disk Utility Modifier Statement (Part 2 of 2)

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

r,(U,n,m),t

U
identifies the unpack operation.

n
is the size of the input field.

m
is the size of the output field.

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running a tape-to-disk program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'182'
//bASSGNbSYS007,X'191'
//bUPSIb11000000 (nonstandard input label
   checking, assume nonstandard user label
   routine has been cataloged as the fifth
   phase of this program in the core image
   library)
//bDLBLbUOUT,'DISKbFILE',67/300
//bEXTENTbSYS007,000123,1,0,00176,00004
//bEXTENTbSYS007,000123,1,1,00860,00040
//bEXECbTPDK
//bUTDbTR,FV,A=(320),B=(600),OY,IN
//bEND
/&
```

# Tape to Printer

The tape-to-printer program can display a tape file in either data display or data list format. Data display provides a byte-for-byte representation of the data file where every byte appears in the listing; this format can handle fixed, variable, and undefined records. Data list provides a simple edited representation of the file; input records must be fixed or variable length and (for the data list function only) must not exceed the size of the print line. The field select option may be used. An option is available to this program to specify the number of logical records in a file to be bypassed before printing begins.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTD,FU,A=(1000),B=(120),IU,OX,PY,R1,S1
```

The format and entries for the utility modifier statement for this program are:

```
//bUTPbTt,Ff,A=(input),B=(output),Ix,Ox,Px,
   Rx,Sx
```

Figure 22 shows detailed information of the entries in the utility modifier statement for the tape-to-printer program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| TPb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file program to transfer fields from an input record to the same or a different relative location of the output record. This is valid only for data list mode. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete in one statement. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the print line.<br><br>/ (slash) separates selected fields. |

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

```
r,(U,n,m),t
```

U
: identifies the unpack operation.

n
: is the size of the input field.

m
: is the size of the output field.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function<br>Tt | TD<br>TL<br>TLF | T | The initial T identifies this as the type of function parameter. |
| | | D | Display |
| | | L | List |
| | | LF | List and Field Select |
| Format<br>Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records.<br>With undefined records, the only valid entry above is data display. |
| Input Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| Output Description | B=(p) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (p) | For printer output, the size of the print line (120, 132, or 144) must be entered.<br>A print line size of 120 is forced if TD is specified. |
| | B=(n,p) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,p) | For field select of variable length records with printer output records, the fixed portion of each output record (the letter n) and the size of the print line (the letter p) must be enclosed in parentheses and separated by a comma. |
| Rewind Input | IR<br>IN<br>IU | I | The first letter in these forms identifies this parameter. The rewind option for the input tape is active both before and after data transfer. |
| | | R | Rewind both before and after data transfer. |
| | | N | Do not rewind either before or after data transfer. |
| | | U | Rewind before and rewind and unload after data transfer. |
| Print Output<br>Ox | OX<br>OC | O | The first letter in these forms identifies this parameter. |
| | | X | Hexadecimal Printout (for data display only) |
| | | C | Character Printout (forced for data list) |
| | | | The type of output indicated by the field-select parameter (hexadecimal or character) overrides this parameter. |

Figure 22. Tape-to-Printer Utility Modifier Statement (Part 1 of 2)

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Spacing Option Sx | S1<br>S2<br>S3<br>SA<br>SB<br>SC<br>SD | S | The first letter in these forms identifies this parameter. |
| | | 1 | Single Spacing (forced for data display) |
| | | 2 | Double Spacing |
| | | 3 | Triple Spacing |
| | | A | Type A First Character Forms Control |
| | | B | Type B First Character Forms Control |
| | | C | Type C First Character Forms Control |
| | | D | Type D First Character Forms Control |
| Page-Numbering Px | PY<br>PN | P | The first letter in these forms identifies this parameter. |
| | | Y | Number pages (forced for data display) |
| | | N | Do not number pages (forced for first character forms control) |
| First Record Rx | Rx | R | The first letter in these forms identifies this parameter. |
| | | x | This represents the position of the first logical record to be printed; x-1 will be bypassed. |

Figure 22. Tape-to-Printer Utility Modifier Statement (Part 2 of 2)

HEXADECIMAL

When a program has printed output, the field selected may be printed in hexadecimal representation. This operation is indicated as follows: r,(X,n),t

X identifies the hexadecimal operation.

n is the size of the input field.

Only the field length of the input is necessary for this operation because the output length will always be assumed to be twice as large. X and n are enclosed in parentheses and separated by a comma.

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

The sample control statement input stream for running a tape-to-printer program from the core image library follows; device and file descriptions are peculiar to the job being run. See page 27 for a description of the heading statement.

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'182'
//bASSGNbSYS005,X'00E'
//bUPSIb10000000 (no label checking)
//bEXECbTPPR
//bUTPbTLF,FV,A=(37,98),B=(40,132),PN,OC,S2
//bFSb1,37,1/CV
//bEND
/&
```

The tape-to-tape program transfers a file from one or more tape reels to one or more other reels. These files may be copied, reblocked, field selected, or reblocked and field selected. If the reblock or field select options are used, the input records must be fixed or variable length.

## Utility Modifier Statement

This statement contains information required for the operation of this program. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbTC,FU,A=(1000),B=(1000),IU,OU,R1
```

The format and entries for the utility modifier statement for this program are:

```
//bUTTbTt,Ff,A=(input),B=(output),Ix,Ox,Rx
```

Figure 23 shows detailed information of the entries in the utility modifier statement for the tape-to-tape program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| TTb | The initials of the program. These initials can be omitted if the statement is used for more than one program. |

## Field Select Statement

This statement provides the information for the file-to-file programs to transfer fields from an input record to the same or a different relative location of the output record. As many field select statements as necessary may be used. If the statement is punched in cards, each card need not be filled even if additional field select cards follow. The field selected must be complete on one card. The format and contents of this statement are:

```
//bFSbr,s,t/r,s,t/r,s,t
```

| Contents | Explanation |
|---|---|
| //bFSb | //b identifies this as a control statement.<br><br>FS identifies this as a field select control statement. |
| r,s,t/ | r indicates the starting position, relative to one (not zero), of the field in the input record to be selected.<br><br>, (comma) separates the entries in the parameter.<br><br>s indicates the length of the field in bytes.<br><br>, separator.<br><br>t indicates the starting position, relative to one, of the output record.<br><br>/ (slash) separates selected fields. |

PACK

When the input field is to be packed before it is placed in the output record, the field select parameter will appear in this form:

```
r,(P,n,m),t
```

P
: identifies the pack operation.

n
: is the size of the input field.

m
: is the size of the output field.

UNPACK

When the input field is to be unpacked before it is placed in the output record, the field select parameter will appear in this form:

```
r,(U,n,m),t
```

U
: identifies the unpack operation.

n
: is the size of the input field.

m
: is the size of the output field.

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Function Tt | TC<br>TF<br>TR<br>TRF | T | The first letter in these forms identifies this parameter. |
| | | C | Copy |
| | | F | Field Select |
| | | R | Reblock |
| | | RF | Reblock and Field Select |
| Format Ff | FF<br>FV<br>FU | F | The leading F of these three possible forms identifies this as the format parameter. |
| | | F | The second F must be indicated for fixed-length records. |
| | | V | The letter V must be indicated for variable-length records. |
| | | U | The letter U must be indicated for undefined records. |
| Input Description | A=(n,m) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (n,m) | For fixed-length input records, the input record length (the letter n) and the input block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | A=(g) | A= | This letter and symbol indicate this is the input-description parameter. |
| | | (g) | For undefined input records or variable input records without field select, the maximum block length must be enclosed in parentheses. |
| Output Description | B=(n,m) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (n,m) | For fixed-length output records, the output record length (the letter n) and the output block length (the letter m) must be enclosed in parentheses and separated by a comma. For field select with variable length records, the letter n indicates the size of the fixed portion of each record, and the letter m indicates the maximum block size. |
| | B=(g) | B= | This letter and symbol indicate this is the output-description parameter. |
| | | (g) | For undefined output records or variable output records without field select, the maximum block length must be enclosed in parentheses. |
| Rewind Option for Input Ix | IR<br>IN<br>IU | I | The first letter in these forms identifies this parameter. The rewind option for the input tape is active both before and after data transfer. |
| | | R | Rewind both before and after data transfer. |
| | | N | Do not rewind either before or after data transfer. |
| | | U | Rewind before and rewind and unload after data transfer. |
| First Record Rx | Rx | R | The first letter in this form identifies this parameter. |
| | | x | This represents the position of the first logical input record to be output (x-1 records will be bypassed). If the file is to be copied, the function parameter must be indicated to be reblocked and the input and output file description parameters must contain identical values. |
| Rewind Output Ox | OR<br>ON<br>OU | O | The first letter in these forms identifies this parameter. The rewind option for the output tape is active both before and after data transfer. |
| | | R | Rewind both before and after data transfer. |
| | | N | Do not rewind either before or after data transfer. |
| | | U | Rewind before and rewind and unload after data transfer. |

Figure 23. Tape-to-Tape Utility Modifier Statement

COPY VARIABLE

When the section of a variable-length record, not defined as the fixed portion, is to be transferred to the output record, the letters CV (copy variable) must be present in the field select statement. If this entry is made when processing records that have been defined as fixed length, an error will be indicated. The CV entry can be entered:

- Before the first field to be selected.
- Between selected fields.
- Following selected fields.

Examples:

```
//bFSbCV/r,s,t/r,s,t
//bFSbr,s,t/CV/r,s,t
//bFSbr,s,t/r,s,t/CV
```

The variable section of the record is placed in the output record following the fixed portion of the record as described in the output description parameter.

## Control Statement Stream

A sample control statement input stream for running a tape-to-tape program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'180',X'A8'
//bASSGNbSYS005,X'181'
//bTLBLbUOUT,'EXAMPLEbFILE',67/300,000123,
   0001,0001,0001,01
//bUPSIb10000000 (unlabeled input and
   standard output labels)
//bEXECbTPTP
//bUTTbTR,FF,A=(100,100),B=(100,1000),OU,
   IR
//bEND
/&
```

# Part 2. Special—Purpose Utility Programs

# Assign Alternate Track—Data Cell

The assign alternate track-data cell program is designed to assign an alternate track on an IBM 2321 Data Cell at any time other than when initializing the data cell. An alternate track can also be unassigned, with the data being reinstated to the primary track. If an alternate track is found defective, you must specify a new alternate to the original primary track. If update records are supplied as input, the program will use them to replace bad records on a specified track.

ASSIGNING AN ALTERNATE TRACK

At the beginning of this program, the format 4 record of the volume table of contents (VTOC) is checked to determine the next alternate track to be assigned. If no alternate track is available, the condition is logged and the job is terminated.

If an alternate track is assigned, the records contained on the defective track are transferred to the alternate track beginning with the data portion of record zero (R0) and continuing with the count, key, and data of R1 through Rn. The record in error is transferred to the alternate track as read. The address of the alternate track is written in the track descriptor record (R0) of the defective track. This address is used as a pointer to the proper alternate track.

Surface analysis is performed on the track in error, if it has been specified, after all records have been transferred to the alternate track. If the error is permanent, this is flagged as a defective track. The track is also flagged as defective if surface analysis is not specified.

If the error is only temporary, the data is transferred back to the originally defective track and the flag byte of the original track is reset to indicate a good track. A record that is written on the alternate track in error will be transferred back in error. The flag byte of the alternate track is set to indicate that this track is unassigned, and the format 4 record is modified to indicate the address of the next available alternate track to be assigned.

If at the completion of surface analysis of the defective track, it is determined that the home address and/or the track descriptor record (R0) area is defective, the track is then flagged defective and points to the alternate track by advancing the home address and R0 800 bytes. If an error occurs at this time, the data remains on the alternate track and the job is terminated.

Effects of Defective Areas

The location of defective areas on the defective track may affect the transfer of records to the alternate track or cause the program to end. The possible location of defective areas and their effect on program processing are:

| Defective Area Location | Effect |
|---|---|
| Address marker of a record or Gap preceding the count area of a record or Count area of record | The record is bypassed (it will not be transferred to the alternate track). |
| Key area of a record or Data area of a record | The record is transferred to the alternate track exactly as it is read. |
| Gap following the count area of a record | The count area is transferred to the alternate track. The remainder of the record (key and data area) filled with A's on the alternate track. |
| Gap following the key area of a record | The data is unrecoverable. |
| Gap between the data area of a record and the address marker of the next record | The records are transferred to the alternate track exactly as they are read. |

Record Printing Option

The program allows the option of printing all records transferred to the alternate track or only those that were read in error from the defective track. You specify your

choice in the utility modifier statement. The records are printed on the device assigned to SYSLST. However, if an error is found in the home address or track descriptor record of the defective track, all records are printed, regardless of the specification in the utiliity modifier statement.

Update Record

The last phase of the program will recognize any number of update records if you specify the update parameter on the utility modifier statement. Each update record must be immediately preceded in the input stream by a track statement, which specifies the location at which the update record should be written. The program writes an update record at the location specified by the preceding track statement, whether or not a record already exists at this location. (If the addition of an update record causes the track capacity to be exceeded, the program writes an error message and then terminates. In this case, the last record on the track may be invalid.)

Update records must appear in hexadecimal representation, two characters per byte, in the input stream. They must include the key and data portions of the record. Do not imbed insignificant characters in an update record: that is, use every column of each input record until all the data has been supplied. Forty bytes of data will fit in one 80-column input record.

## Utility Modifier Statement

This statement contains information required for the operation of this program. The first entry R=(cccchhh) must be supplied and the following parameters are assumed:

```
//bUbR=(cccchhh),OY,IF,UN
```

The format and entries for the utility modifier statement for this program are:

```
//bUATbR=(cccchhh),Ox,Ix,Ux
```

| Entry | Reason |
|---|---|
| //bU | These entries identify the utility modifier statement. |
| ATb | These letters indicate that this is the alternate track assignment program. These entries can be omitted. |
| R= | This letter and symbol indicate this is the track location parameter. |
| ($c_1c_2c_2c_3h_4h_5h_5$) | This entry indicates the track to which an alternate or a new alternate track is to be assigned, or a track that is to have its condition flag changed from defective to nondefective.<br><br>$c_1$=cell (0-9)<br>$c_2c_2$=subcell (00-19)<br>$c_3$=strip (0-9)<br>$h_4$=cylinder (0-4)<br>$h_5h_5$=track (00-19)<br><br>These entries must be enclosed in parentheses. Note that only strip 0-5 may be specified with subcell 19. |
| Ox | Identifies this parameter as the output option parameter.<br><br>x=Y indicates that all records are to be printed as they are transferred to the alternate track.<br><br>x=N indicates that only the record or records found to be in error are to be printed. |
| Ix | Identifies this parameter as the input option parameter.<br><br>x=F indicates that an alternate track will be assigned without performing a surface analysis. This is the assumed option when the Ix parameter is omitted.<br><br>x=A indicates that surface analysis should be performed on the specified track (or on an alternate, if one is already assigned) and, if the error is permanent, an alternate track will be assigned.<br><br>x=U indicates that the flag from a defective track is to be unconditionally removed |

| | |
|---|---|
| | and the data recopied onto the primary track. Any error in the transfer of data will be indicated to the operator. |
| Ux | Identifies this parameter as the update parameter.<br><br>x=Y indicates that one or more update records will follow in the input stream.<br><br>x=N indicates that no update records will follow in the input stream. |

## Track Statement

The format and entries for the track statement for this program are:

TRACK=ccccchhhhrrkkdddd[S]

| Entry | Reason |
|---|---|
| TRACK= | This entry indicates that this is the record location parameter. |
| $c_1c_1c_2c_2h_1h_1h_2h_2$rrkkdddd[S] | This entry specifies the track location and record description necessary for the utility to write the update record. This information must be specified in hexadecimal representation.<br><br>$c_1c_1$=subcell (00-13).<br>$c_2c_2$=strip (00-09).<br>$h_1h_1$=cylinder (00-04).<br>$h_2h_2$=track (00-13).<br>rr=record number (00-FF).<br>kk=key length (00-FF)<br>dddd=data length (0000-FFFF).<br><br>S must appear if the record is to be written using the write special count, key and data command. |

## Control Statement Stream

A sample control statement input stream for running an assign alternate track-data cell program follows:

```
//bJOBbALTRK
//bASSGNbSYS000,X'293'
//bEXECbATAM
//bUATbR=(2050212),OY,IA
/&
```

A sample control statement input stream to use this program to update a record follows:

```
//bJOBbUPDATE
//bASSGNbSYS000,X'293'
//bEXECbATAM
//bUATbR=(2050212),OY,IA,UY
TRACK=0500020C04000050
    (first 40 bytes of update record)
                    .
                    .
                    .
    (last 40 bytes of update record)
/*
/&
```

# Assign Alternate Track—Disk

Use the assign alternate track-disk program with the IBM 2311 Disk Storage Drive, the IBM 2314 Direct Access Storage Facility, or the IBM 2319 Disk Storage. The program functions are to assign an alternate track to a defective track, to change the flag indicating the track's condition from defective to nondefective, and if update records are supplied as input, to replace bad records on a track. Also, if an alternate track is found defective, you must specify a new alternate to the original primary track. Specifications contained in the utility modifier statement indicate which function the program is to perform.

## ASSIGNING AN ALTERNATE TRACK

The defective track is identified by specifications in the utility modifier statement. The alternate track is identified by the format 4 record of the volume table of contents (VTOC) on the disk pack. The records from the defective track are transferred to the alternate track, beginning with the data area of the track descriptor record (R0). The VTOC format 4 record is modified to contain the address of the next available alternate track. The condition of the defective track may then be analyzed, depending upon your specifications in the utility modifier statement.

When the condition of the track is not analyzed, or when it is analyzed and found to be defective, the track is flagged defective and the track descriptor record is modified to contain the address of the alternate track. When the condition of the track is analyzed and found to be nondefective, the track is flagged nondefective, the records are transferred back to the track from the alternate track, and the VTOC format 4 record is reset to indicate that the alternate track is still available for assignment.

The program is complete when an alternate track has been assigned or when the defective track has been found nondefective and flagged accordingly.

### Effects of Defective Areas

The location of defective areas on the defective track may affect the transfer of of records to the alternate track or cause the program to end. The possible location of defective areas and their effect on program processing are:

| Defective Area Location | Effect |
|---|---|
| Address marker of a record<br>or<br>Gap preceding the count area of a record<br>or<br>Count area of a record | The record is bypassed (it will not be transferred to the alternate track). |
| Key area of a record<br>or<br>Data area of a record | The record is transferred to the alternate track exactly as it is read. |
| Gap following the count area of a record | The count area is transferred to the alternate track. The remainder of the record (key area and data area) on the alternate track is filled with A's. |
| Gap following the key area of a record | The count area and key area of the record are transferred to the alternate track. The data area of the record on the alternate track is filled with A's. |
| Gap between the data area of a record and the address marker of the next record | The records are transferred to the alternate track exactly as they are read. |
| Home address of the track<br>or<br>Track descriptor record (R0) | IBM 2311: The program is ended (the track cannot be flagged defective).<br><br>IBM 2314/2319: The gap between the track index marker and home address is extended by 705 bytes and an attempt is made to |

write the home address and track descriptor record following the extended gap. If the home address or track descriptor record cannot be written, the program is ended (the track cannot be flagged defective).

Record Printing Option

The program allows the option of printing all records transferred to the alternate track or only those that were read in error from the defective track. You specify your choice in the utility modifier statement. The records are printed on the device assigned to SYSLST. However, if an error is found in the home address or track descriptor record of the defective track, all records are printed regardless of the specification in the utility modifier statement.

CHANGING TRACK-CONDITION INDICATION

The program performs this function for tracks that have been previously flagged defective and to which alternate tracks have been assigned. The track flagged defective is specified in the utility modifier statement. The alternate track is identified in the track descriptor record of the specified track.

The program flags are specified track nondefective without analyzing the condition of the track. The records from the alternate track are transferred back to the specified track and the address of the alternate track is removed from the track descriptor record of the specified track. (Any error encountered in the transfer is indicated to the operator.) The flag byte on the alternate track is changed to indicate that the alternate track is not assigned. If this alternate track was the last alternate track assigned, the VTOC format 4 record is changed to indicate that this track is not the next alternate track available for assignment. Otherwise, the VTOC format 4 record is not changed and the alternate track cannot be assigned using this program.

Update Record

The last phase of the program will recognize any number of update records if you specify the update parameter on the utility modifier statement. Each update record must be immediately preceded in the input stream by a track statement, which specifies the location at which the update record should be written. The program writes an update record at the location specified by the preceding track statement, whether or not a record already exists at this location. (If the track capacity to be exceeded, the program writes an error message and then terminates. In this case, the last record on the track may be invalid.)

The track to which an alternate has been assigned and the track stated in the TRACK= statement must be the same.

Update records must appear in hexadecimal representation, two characters per byte, in the input stream. They must include the key and data portions of the record. Do not imbed insignificant characters in an update record: that is, use every column of each input record until all the data has been supplied. Forty bytes of data will fit in one 80-column input record.

## Utility Modifier Statement

This statement contains information required for the operation of this program. The first entry R=(ccccchhh) must be supplied and the following parameters are assumed:

```
//bUbR=(ccccchhh),OY,IF,UN
```

The format and entries for the utility modifier statement for this program are:

```
//bUATbR=(ccccchhh),Ox,Ix,Cn,Ux
```

Figure 24 shows detailed information of the entries in the utility modifier statement for the assign alternate track-disk program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| ATb | The initials of the program. These initials can be omitted. |

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Track Location | R=(ccccchhh) | R= | This letter and symbol indicate this is the track location parameter. |
| | | (ccccchhh) | This entry indicates the track to which an alternate or a new alternate track is to be assigned, or which is to have its condition flag changed from defective to nondefective (cccc denotes the cylinder number -- cylinders 0-199 only -- and hhh denotes the head number). These entries must be enclosed in parentheses. |
| Output Option<br><br>Ox | OY<br>ON | O<br><br>Y<br><br>N | The first letter in these forms identifies this as the output option parameter.<br><br>All records are to be printed as they are transferred to the alternate track.<br><br>Only the record(s) found to be in error is to be printed. |
| Input Option<br><br>Ix | IF<br>IA<br>IU | I<br><br>F<br><br>A<br><br>U | The first letter in these forms identifies this as the input option parameter.<br><br>The specified track should be flagged without surface analysis.<br><br>Surface analysis should be performed on the specified track if the track has not been previously flagged. If the primary track has been flagged, then the alternate track assigned will have surface analysis performed on it.<br><br>The flag from a defective track should be unconditionally removed and the data recopied onto the prime track. Any error in the transfer of data will be indicated to the operator. |
| Count Option<br><br>Cn | Cn | C<br><br>n | The first letter in this form identifies this as the count option parameter.<br><br>A decimal value of 1 to 255 should be entered indicating the number of times surface analysis is to be performed. This parameter is ignored if the input parameter is IF or IU. Enlarging this parameter causes a proportional increase in program run time. |
| Update Option Ux | UY<br>UN | U<br><br>Y<br><br>N | The first letter in these forms identifies this as the update option parameter.<br><br>One or more update records will follow in the input stream.<br><br>No update records will follow in the input stream. |

Figure 24. Assign Alternate Track-Disk Utility Modifier Statement

## Track Statement

The format and entries for the track statement for this program are:

```
TRACK=cccchhhhrrkkdddd[S]
```

| Entry | Reason |
|---|---|
| TRACK= | This entry indicates that this is the record location parameter. This entry must refer to the primary track to which an alternate has been assigned. |
| c c c c h h h h rrkkdddd[S] | This entry specifies the track location and record description necessary for the utility to write the update record. This information must be specified in hexadecimal representation.<br><br>c c c c =cylinder number.<br>h h h h =track number.<br>rr=record number (00-FF).<br>kk=key length (00-FF).<br>dddd=data length (0000-FFFF).<br><br>S must appear if the record is to be written using the write special count, key and data command. |

## Control Statement Stream

A sample control statement input stream for running an assign alternate track-disk program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS000,X'191'
//bEXECbATAD
//bUATbR=(0027003),IA,C3,CY
/&
```

A sample control statement input stream to update two records follows:

```
//bJOBbUPDATE
//ASSGNbSYS000,X'192'
//bEXECbATAD
//bUATbR=(0004002),IA,C3,CY,UY
TRACK=00040002030000 50
       (first 40 bytes of update record)
       (last 40 bytes of update record)
TRACK=00040002040001E0
       (first 40 bytes of update record)
                    .
                    .
                    .
       (last 40 bytes of update record)
/*
/&
```

# Clear Data Cell

The clear-data cell program clears one or more areas of the IBM 2321 Data Cell Drive, and establishes preformatted tracks containing an indicated base throughout the area cleared. The control information for the operation of this program is entered in three types of control statements.

The first type of control statement (job control) defines channel and unit assignment, physical-device description, and areas of the data cell to be processed. The second type of control statement (utility modifier) contains the information unique to this program. The third type of statement is an END card.

The area to be cleared can be as small as one track or up to a maximum of a complete data cell. Any number of areas can be designated to be cleared with one run of this program. When an area of data cell is cleared, fixed-length blocks containing count, key, and data areas are established on the data cell. The information defining the key and data areas is indicated in the utility modifier statement, or, if a utility modifier statement is not entered, values are assumed. The count area is generated with:

Cylinder number (2 bytes)
Head number (2 bytes)
Record number (1 byte)
Key length (1 byte)
Data length (2 bytes).

The key and data areas defined, with the exception of the first eight bytes of the data portion of the track descriptor record (R0), are filled with a user-defined character. The first eight bytes of the data portion of the track descriptor record (R0) are written:

Bytes 1-2 - Cylinder number

Bytes 3-4 - Head number

Byte 5 - Record number (always zero)

Bytes 6-7 - Number of unused bytes on the track

Byte 8 - Binary zero.

Label checking determines whether the area to be cleared contains all or part of an unexpired file. Expired labels for the area to be cleared are deleted from the VTOC (volume table of contents).

## Utility Modifier Statement

This statement allows three parameter entries. The first parameter defines the length of the key and data block. The second parameter defines the fill character. The third parameter allows the option to write data cell check or not write data cell check. The format and entries for this parameter are:

```
//bUCMbB=(K=l,D=l),{C'c' },O{Y},E=(e)
                   {X'xx'}  {N}
```

If the utility modifier statement is omitted, the assumed values are:

```
//bUCMbB=(K=0,D=100),X'00',OY,E=(2321)
```

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| CMb | The initials of the program. These initials can be omitted. |

| Parameter | Entry | Explanation |
|---|---|---|
| B=(K=l,D=l) | B= | Identifies this parameter. |
| | (K=l, D=l) | Indicates the length of the key and data block in bytes. If a key length is not desired, the key length must equal zero. |
| C'c' | C'c' | C is entered and followed by the fill character (EBCDIC) enclosed in apostrophes. |
| X'xx' | X'xx' | X is entered and followed by the hexadecimal fill character enclosed in apostrophes. |
| OY | O | Identifies this as the output parameter. |
| cr | | |
| ON | Y | Indicates write data cell check (forced for this program). |
| | N | Indicates do not write data cell check (ignored for this program). |

| Parameter | Entry | Explanation |
|---|---|---|
| E=(e) | E= | Indicates this is the device type description parameter. |
| | (e) | For output devices, the valid entry is 2321. This entry must be enclosed in parentheses. |

## END Statement

This must be the last control statement. The statement is entered:

```
//bEND
```

| Entry | Reason |
|---|---|
| //b | These entries identify this as a control statement. |
| END | This entry indicates that this is the last control statement. |

This program is resident in the core image library in three phases:

```
CLDC
CLDC2
CLDC3
```

## Control Statement Stream

A sample ccntrol statement input stream for running a clear data cell program from the core image library follows; device and extent descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS008,X'193'
//bDLBLbUOUT,'FILEbLABEL',67/300
//bEXTENTbB=4,SYS008,000012,1,0,05000,00100
//bEXECbCLDC
//bUCMbB=(K=0,D=900),C'X'
//bEND
/&
```

# Clear Disk

The clear-disk program clears one or more areas of disk on the IBM 2311 Disk Storage Drive, the IBM 2314 Direct Access Storage Facility, or the IBM 2319 Disk Storage, and establishes preformatted tracks containing an indicated base throughout the area cleared. Both types of disks cannot be cleared in the same job. The control information for the operation of this program is entered in three types of control statements.

The first type of control statement (job control) defines channel and unit assignment, physical-device description, and areas of disk to be processed. The second type of control statement (utility modifier) contains the information unique to this program. The third type of statement is an END card.

The area to be cleared can be as small as one track or up to a maximum of a complete disk pack. Any number of areas can be designated to be cleared with one run of this program. When an area of disk is cleared, fixed-length blocks containing count, key, and data areas are established on the disk. The information defining the key and data areas is indicated in the utility modifier statement, or, if a utility modifier statement is not entered, values are assumed. The count area is generated with:

Cylinder number (2 bytes)

Head number (2 bytes)

Record number (1 byte)

Key length (1 byte)

Data length (2 bytes).

The key and data areas defined, with the exception of the first eight bytes of the data portion of the track descriptor record (R0), are filled with a user-defined character. The first eight bytes of the data portion of the track descriptor record (R0) are written:

Bytes 1-2 - Cylinder number

Bytes 3-4 - Head number

Byte 5 - Record number (always zero)

Bytes 6-7 - Number of unused bytes on the track

Byte 8 - Binary zero.

This program requires that the DLBL card indicate a sequential file. Label checking determines whether the area to be cleared contains all or part of an unexpired file. Expired labels for the area to be cleared are deleted from the VTOC (volume table of contents).

## Utility Modifier Statement

This statement allows three parameter entries. The first parameter defines the length of the key and data block. The second parameter defines the fill character. The third parameter allows the option to write disk check or not to write disk check. The format and entries for this parameter are:

```
//bUCLbB=(K=l,D=l),{C'c' },O{Y},E=(e)
                   {X'xx'}  {N}
```

If the utility modifier statement is omitted, the assumed values are:

```
//bUCLbB=(K=0,D=100),X'00',OY,E=(2311)
```

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| CLb | The initials of the program. These initials can be omitted. |

| Parameter | Entry | Explanation |
|---|---|---|
| B=(K=l,D=l) | B= | Identifies this parameter. |
| | (K=l, D=l) | Indicates the length of the key and data block in bytes. If a key length is not desired, the key length must be zero. |
| C'c' | C'c' | C is entered and followed by the fill character (EECDIC) |

| | | |
|---|---|---|
| | | enclosed in apostrophes. |
| or | | |
| X'xx' | X'xx' | X is entered and followed by the hexadecimal fill character enclosed in apostrophes. |
| OY | O | Identifies this as the output parameter. |
| or | | |
| ON | Y | Indicates write disk check. |
| | N | Indicates do not write disk check. |
| E=(e) | E= | Indicates this is the device type description parameter. |
| | (e) | For output devices, the valid entries are 2311 and 2314. This entry must be enclosed in parentheses. |

## END Statement

This must be the last control statement. The statement is entered:

```
//bEND
```

| Entry | Reason |
|---|---|
| //b | These entries identify this as a control statement. |
| END | This entry indicates that this is the last control statement. |

This program is resident in the core image library in three phases:

```
CLRDSK
CLRD2
CLRD3
```

## Control Statement Stream

A sample control statement input stream for running a clear disk program from the core image library follows; device and extent descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS012,X'191'
//bDLBLbUOUT,'DISKbLABEL',67/300
//bEXTENTbSYS012,001221,1,0,00310,00630
//bEXECbCLRDSK
//bUCLbB=(K=38,D=480),X'55',CN
//bEND
/&
```

Note: When clearing cylinder 200 of a disk pack to be used by the System/360 Model 30 or Model 40 Emulator (program numbers EU484, EU485), or by the System/370 Emulators of 1401, 1440, 146C, and 1410/7010 programs (program number EU490), you should include immediately before the EXEC card in the control statement stream an UPSI card of the following format:

```
//bUPSIb00000001
```

This card will cause cylinder 200 to be cleared along with the area specified on the EXTENT card. Cylinder 200 should not be specified on the EXTENT card. If only part of the pack is to be cleared and not cylinder 200, do not include this UPSI card.

# Copy and Restore Disk or Data Cell

Disk operating system volumes or files may be transferred from disk to cards, disk, or tape, or from data cell to tape by means of these programs. Data can be restored to a disk pack or data cell at a later date. Applicable devices can be the IBM 2311, 2314, or 2321; however, the same device type must be used when restoring data. This transfer processes all fields (that is, data portion of R0 and count, key, and data portions of records R1 through Rn) necessary to restore the data so that it is identical to the original volume or file. The output created by the copy programs, except copy disk to disk, is designed for restore program use only. The copy and restore programs are:

- Copy disk to card and restore card to disk.
- Copy disk or data cell to tape and restore tape to a device of the original type.
- Copy disk to disk.

CHECKPOINT AND RESTART FACILITY

The copy and restore programs provide checkpoint and restart facilities through the DOS system. (Refer to the DOS System Control and Service publication listed on the front cover of this publication.)

Checkpoint information may be taken at the beginning of every 80th track or at the beginning of each new extent. This information must be written on a disk or tape separate from the input and/or output device for the particular job. SYS003 is the programmer unit for assigning the checkpoint file. If SYS003 is not assigned to a tape, disk, or data cell, the checkpoint facility will be ignored.

Note: When checkpoints are not desired, you must ensure that SYS003 is unassigned by using job control card:

```
//bASSGNbSYS003,UA
```

The size of the checkpoint records taken depends on the record size specified in the A parameter of the utility modifier card, and the amount of main storage available.

| Device | Record Size (in bytes) | Problem Program Area | Size of Checkpoint Record |
|---|---|---|---|
| 2311 | Up to 3625 | 10K | 7,400 bytes |
| 2314/ 2319 | Up to 6400 | 10K | 10,100 bytes |
| 2311 | Up to 3625 | 12K | 11,100 bytes |
| 2314/ 2319 | Up to 7294 | 12K | 11,100 bytes |
| 2314/ 2319 | Up to 7294 | 18K | 18,400 bytes |
| 2321 | Up to 2000 | 10K | 7,900 bytes |

If checkpoints are written on disk, refer to "Checkpoints on Disk" in the DOS Supervisor and I/O Macros publication listed in the preface of this publication for further information.

Before a checkpoint is taken during the copy disk to card program, this message is printed:

```
CHECKPOINT BEING TAKEN FOLLOWING CARD NO.
xxxxxx
```

The DOS system then prints a message identifying the checkpoint number. All cards following the card referenced in the first message should be deleted before attempting to restart. Similarly, if restoring from card to disk when the checkpoint is taken, all cards following the card referenced in the first message should be reloaded in the card reader.

The filename associated with the checkpoint file is UCHKPT. This filename is to be used in the TLBL and DLBL cards whenever standard labels are to be processed. It must also be used in the RSTRT card when the checkpoints are on disk.

## Copy Disk or Data Cell Programs

The copy programs can transfer data from disk to card, disk, or tape, or from data cell to tape in either a copy volume or copy file.

The copy volume function transfers data from one disk pack to cards, disk, or tape, or from one data cell to tape. The programmer unit is SYS004. The complete volume is transferred, including the two

IPL records, the volume label(s), and the VTOC.

The copy file function transfers one data file from disk to cards, disk, or tape, or from data cell to tape. You must supply the DLBL and EXTENT cards to describe the file to be copied. The extent limits should match the corresponding extents in the VTOC (rather than the extents specified in creating the file, in the event that not all of the specified extents are needed). If the supplied extent limits do not match the corresponding extents in the VTOC, the file described by the job control statements will be copied without an error indication. The file is transferred to the same extent limits as it was copied from. The extent limits specified in the EXTENT-cards following the UOUT statement are checked for unexpired files. The copy file function can also be used for a multiple-extent sequential disk file on more than one pack if you ensure that each extent is assigned.

If either copy function is used, the count fields may not be duplicated on any track, and the volume must be initialized according to IBM standard (i.e., standard home address, R0, volume label(s), etc.). See "Initialize Disk" or "Initialize Data Cell." Consecutive, indexed-sequential, and direct access method file organizations are supported by this group of programs.

## I/O AREA

I/O overlap may be possible if sufficient storage is available to process two or more complete tracks at one time and as channel assignments permit.

Note for Model 30: When tape and disk have been assigned to different kinds of channels (i.e. multiplexor and selector channels), it is necessary to assign one input/output area by using the O parameter. This prevents tape overrun errors.

## OPENING A DISK PACK OR DATA CELL

Standard label checking is performed by using the system OPEN for the copy file function. For the copy-disk-to-disk program, refer to "Opening a Disk Pack or Data Cell for Restoring a File". For information about the output file, refer to "Opening a Disk Pack or Data Cell for Restoring a Volume."

## PROCESSING USER STANDARD LABELS ON DASD FILES

User standard labels on consecutive and direct address files on a disk pack or data cell can be copied by the copy file function if an UPSI job control card is supplied. User standard labels are copied only if you assign the value 1 to bit 7 of the UPSI byte. Bits 0 and 2 have significance only for tape label processing (see "Opening the Tape Volume" below). All other bits of the UPSI byte have no significance to the copy disk or data cell programs. If no UPSI card is supplied, it is assumed that there are no user standard labels present or that those present are not to be copied and restored. When a file that was copied with a // UPSI 00000001 card in the job stream is subsequently restored, you must reset bit 7 to 1 (on) to ensure that the restore extents are identical to the copy extents.

## OPENING THE TAPE VOLUME

Tape files containing either no labels or IBM standard labels can be processed. Nonstandard labels are not supported by the copy/restore programs. When any label processing is to be performed, the UPSI job control card must set bits 0 and 2 as follows (0 equals off, 1 equals on):

Bit 0 Off = standard input-label checking.
On = no input-label checking.

Bit 2 Off = standard output-label checking.
On = no output-label checking.

If standard label checking is desired:

1. An UPSI card is optional (the copy/restore programs assume standard label checking).
2. A TLBL card is required.

If no label checking is desired:

1. An UPSI card is required.
2. A TLBL card is not required.

## UTILITY MODIFIER STATEMENT

This statement must be supplied to the copy program for program operation. The parameters are free form in that they can be punched in any order. The entries and description of the utility modifier statement are:

```
//bUCRbTt,A=(a),CELLS=(n,n,n,n,n),N=(n),
   Ox,IPL,E=(e),Mx
```

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |

CRb — These entries are the initials of the program.

Tt — These entries identify the type of function:

TF= copy file
TV= copy volume.

A=(a) — These entries indicate the most common physical record length of the area to be copied. A difference between this length (a) and the actual record length, can decrease performance. If performance is not an important factor, any valid record length will apply.

CELLS= — These entries identify the parameter. The CELLS parameter is used for only those operations involving data cells.

(n,n,n,n,n) — Here each n is a decimal digit from 0-9 designating a cell number. From one to five cell numbers can be included in this parameter. The order in which the cell numbers are included in this parameter is the order in which the cells will be processed by the program.

N=(n) — These entries are required for the copy file function (if n is greater than 1) to indicate the number of volumes for the file. n must not exceed ten (decimal). This parameter is optional for the copy volume function. If used, n must equal one; if not used, n is assumed to be equal to one.

Ox {O1 / O2} — O is the I/O area parameter. This parameter can only be used for the disk/data cell to tape program.

When this parameter is omitted, or O2 is specified, the program determines the I/O area assignments. When sufficient storage is available, two I/O areas are assigned to provide for I/O overlap.

When O1 is specified, the program assigns one I/O area, regardless of available storage. No overlap in the input/output operations occurs. Refer to the note for Model 30 under "I/O Area."

IPL — This parameter is optional for the copy file function, but invalid for the copy volume function. If present, the IPL records are copied and restored.

E=(e) — This is the device type description parameter. The valid entries for the letter e are 2311, 2314 (used for both the 2314 and 2319), and 2321. This entry must be enclosed in parentheses. If this entry is omitted, the assumed value is 2311.
For copy volume, the device type is generated from the physical unit block (PUB) table. In that case, the E=(e) parameter is ignored.

Mx {MS / MD / MI} — M is the file management parameter. This parameter is used only for the copy file function.

MS indicates sequential file management. This is the assumed value if this parameter is omitted.

MD indicates direct access file management.

MI indicates indexed-sequential file management. All extent limits must then be specified for the entire indexed-sequential file (i.e. master index, cylinder index, and overflow areas).

INPUT PROCESSING

The following should be considered to determine program efficiency for any data file. The program determines:

1. The number of physical records that can be read into main storage at a time. This is determined from the specified A parameter and the available I/O area.

2. The number of times that chaining of records may be performed per track.

The program reads physical records (chained) as determined in step 1. When chaining can no longer be performed on the remainder of a track (as determined in step 2), the program reads one physical record at a time until all records have

been read from the track.

If a record overflow condition occurs, control is given to the basic routine (always in main storage) that reads one record per revolution until all records have been read from the track. Chaining will again be used on the next track.

If a record is read that exceeds the available I/O area, the size of the actual record is retained so that when restoring the record, the record maintains its original format. When restoring, the section of the record that could not be read is then replaced with binary zeros. A message will identify this condition during the copy program.

Performance may be reduced when tracks are not filled to maximum capacity. This may occur, for example, when the direct access method of file organization is used.

When using the copy disk or data cell to tape program, each tape reel will be rewound and unloaded when processing is complete for that tape.

## Restore Disk or Data Cell Programs

The restore programs have the ability to restore data from cards or tape to disk, or from tape to data cell (depending on the method used in the corresponding copy program) in either restore volume or restore file.

The restore volume function transfers one volume of data from tape or cards to a disk pack, or from tape to a data cell. The IPL records, volume label(s), and VTOC are automatically restored along with the data. Both restore programs modify those fields in the format 4 label that are exclusively unique to the volume (that is, alternate track information).

The restore file function completes the transfer of data so that the restored file is identical in location and format to the file at the time the copy function was performed.

Before running the restore program, the applicable disk pack or data cell must be initialized according to IBM standards (i.e., standard home address, R0, volume label(s), etc.). See "Initialize Disk" or "Initialize Data Cell," whichever is applicable.

Specific parameters unique to the program are not needed because the output created by the copy programs contains all the utility control information necessary to execute the corresponding restore program.

The disk or data cell output processing is dependent upon the processing performed in the corresponding copy program; i.e., when restoring, a partition of greater capacity will not improve performance.

### I/O AREA

The exact I/O area assignment made in the corresponding copy program will be assigned for restoring.

### OPENING A DISK PACK OR DATA CELL FOR RESTORING A FILE

The restore programs perform standard label checking by using the system OPEN. You must supply the DLBL and EXTENT cards to be used to create the label for the file that is to be restored. The extent limits must be the same as the limits supplied to the copy program.

If user standard label processing was indicated when copying, bit 7 of the UPSI byte must again be set to 1 when restoring the file.

When more than one data cell volume is to be processed, the program will accept each volume as specified in the CELLS parameter of the copy utility modifier statement. When opening, and the last assigned cell has been processed but additional volumes are required, the program will return to the first cell given in the CELLS parameter. The cycle will be repeated as often as necessary to complete the file.

### OPENING A DISK PACK OR DATA CELL FOR RESTORING A VOLUME

The system opens a disk pack or data cell as a sequential file by means of the restore program (see the Supervisor and I/O Macros publications listed in the preface of this publication). The DLBL and EXTENT information are used to determine the area of the volume to be searched for unexpired labels. This information must indicate a sequential file. The filename on the DLBL card must be UOUT. The extent provided through job control for a restore volume should be as large as possible but must not include track zero on cylinder zero, or the VTOC. An unexpired file that overlaps this extent may be deleted only by choice.

The new VTOC is automatically restored from tape to data cell, or from tape or cards to disk. The program also modifies the following fields, which are dependent

upon the physical volume:

- Format 4 label, field 5 (highest alternate track) is modified to reflect the highest alternate track assigned on the volume.
- Format 4 label, field 6 (number of alternate tracks) is modified to reflect the number of alternate tracks available on this volume.

If user standard label processing was indicated when copying, bit 7 of the UPSI byte must again be set to 1 when restoring the file.

OPENING THE TAPE VOLUME(S)

Tape files containing either no labels or IBM standard labels can be processed. Nonstandard labels are not supported by the copy/restore programs. When any label processing is to be performed, the UPSI job control card must set bits 0, 2, and 7 as follows (0 equals off, 1 equals on):

```
Bit 0  Off = standard input-label checking.
       On  = no input-label checking.

Bit 2  Off = standard output-label
             checking.
       On  = no output-label checking.

Bit 7  Off = user standard labels.
       On  = restoring user standard labels
             (valid for tape file from copy
             disk or data cell program
             only).
```

If standard label checking is desired:

1. An UPSI card is optional (the copy/restore programs assume standard label checking).
2. A TLBL card is required.

If no label checking is desired:

1. An UPSI card is required.
2. A TLBL card is not required.

If label processing is performed when copying onto the tape, label processing must be performed when restoring.

CLOSING THE TAPE VOLUME(S)

Each tape reel is rewound and unloaded when processing is complete for that tape. When an alternate tape drive is assigned, the program will alternate between the primary drive and the alternate drive until processing is complete for the program. The restore program will request the operator to decide at the end of each unlabeled tape whether the tape is at the end of file or volume.

## Control Statement Stream

A sample control statement input stream for running a copy disk to tape program (copy volume function) from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'191'
//bASSGNbSYS005,X'180'
//bASSGNbSYS003,UA (checkpoints not
   desired)
//bUPSIb00100001 (copy user standard
   labels)
//bEXECbCRDT
//bUCRbTV,A=(1600)
/&
```

To restore this same volume from tape to disk, use the following job stream:

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'180'
//bASSGNbSYS005,X'191'
//bASSGNbSYS003,UA (checkpoints not
   desired)
//bDLBLbUOUT
//bEXTENTbSYS005,111111,1,0,C001C,01990
//bUPSIb10000001 (restore user standard
   labels)
//bEXECbCRTD
/&
```

To run a copy disk to card (copy file function) program, use the following job stream:

```
//bJOBbEXAMPLE
//bASSGNbSYS005,X'191'
//bASSGNbSYS006,X'00D'
//bASSGNbSYS003,UA (checkpoints not
   desired)
//bDLBLbUIN,'DISKbFILE',67/100
//bEXTENTbSYS005,111111,1,0,00011,00050
//bEXECbCRDC
//bUCRbTF,A=(1600)
/&
```

To run a copy disk to disk program with indexed sequential files (copy file function), use the following input stream:

```
//bJOBbCOPY
//bASSGNbSYS006,X'191'
//bASSGNbSYS005,X'192'
//bASSGNbSYS003,UA
//bDLBLbUIN,'INDEXED SEQUENTIAL',,ISE (for
   copying)
//bEXTENTbSYS006,333333,4,1,10,2 (cylinder
   index)
//bEXTENTbSYS006,333333,1,2,20,30 (prime
   data area)
//bEXTENTbSYS006,333333,2,3,12,8 (overflow
   area)
//bDLBLbUOUT,'INDEXED SEQUENTIAL',,ISC (for
   restoring)
```

```
//bEXTENTbSYS005,222222,4,1,10,2
//bEXTENTbSYS005,222222,1,2,20,30
//bEXTENTbSYS005,222222,2,3,12,8
//bEXECbCRDD
//bUCRbTF,A=(800),MI
/&
```

To run a copy disk to disk program (copy volume function), use the following stream:

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'191'
//bASSGNbSYS005,X'192'
//bASSGNbSYS003,UA (checkpoints not
                    desired)
//bDLBLbUOUT
//bEXTENTbSYS005,222222,1,0,00010,01990
//bEXECbCRDD
//bUCRbTV,A=(1600)
/&
```

To run a copy data cell to tape program (copy volume function), use the following job stream:

```
//bJOBbCOPY
//bASSGNbSYS003,X'283'
//bTIBLbUCHKPT
//bASSGNbSYS004,X'293'
//bASSGNbSYS005,X'282'
//bTLBLbUOUT
//bUPSIb10100
//bEXECbCRDT
//bUCRbTV,E=(2321),A=(80),CELLS=(5)
/&
```

# Initialize Data Cell

The initialize data cell program prepares from one to five cells for use on an IBM 2321 Data Cell Drive. The preparation of each of these data cells consists of:

- VTOC label checking
- Home address generation
- Track descriptor (R0) generation
- Surface analysis and initialization verification
- Volume label creation
- IPL and VTOC format creation.

## VTOC Label Checking

Before a cell is initialized, a check is performed to see if a volume table of contents (VTOC) is present. If the cell has been previously initialized and the VTOC is present, any labels in the VTOC are checked to see if the files on the data cell have expired. If any files have not expired, a message is printed. If the operator still wishes to bypass the condition, that cell may be deleted from the job.

## Home Address Generation

A binary, five-byte home address is written on every track by this program. The five bytes written are: flag, subcell number, strip number, cylinder number, and track number.

## Track Descriptor (R0) Generation

The track descriptor record is the first record following the home address. This record consists of an 8-byte count portion and an 8-byte data portion.

If an error occurs in the home address or R0 area, the track is flagged defective, and an alternate track is assigned by advancing the home address and R0 records by 800 bytes. If an error occurs at this time, the program is terminated.

## Surface Analysis and Initialization Verification

Surface analysis is performed first in the alternate track area. When a track in the alternate area is found to be defective, the track is flagged defective and cannot be assigned as an alternate track. Surface analysis is then performed on all the remaining tracks on the cell. If a track is detected that has a defective surface area upon which data cannot be written, an alternate track is established. Messages are printed on SYSLOG to notify you of defective tracks, assigned alternate tracks, and their locations. A cell is deleted from further processing when no more alternate tracks are available for assignment.

## Volume Label Creation

Using a control card unique to this program, a volume label is created in the standard format (VOL1) for each cell processed. The volume label is written on subcell 0, strip 0, cylinder 0, track 0, and record 3 of each cell. Seven additional user volume labels (VOL2-VOL8) can be created, if desired, and will be placed in records four through ten.

## IPL and VTOC Format Creation

IPL FORMAT CREATION

This program formats two IPL records. These records are written on subcell 0, strip 0, cylinder 0, track 0, records one and two. Record one is written with a 24-byte data field of binary zeros. Record two is written with a 144-byte data field of binary zeros.

VTOC FORMAT CREATION

The initialize data cell program preformats the VTOC. The location on the cell in which the VTOC is to be placed is indicated in a control card. The standard location of the VTOC is on subcell 0, strip 0, cylinder 0, immediately following the volume labels and extending to the end of the cylinder. However, the VTOC can appear anywhere on the cell (except for the alternate track area), but cannot exceed cylinder boundaries.

## Utility Modifier Statement

This statement must be provided for program operation. The format and entries for the utility modifier statement for this program are:

```
//bUIMbCELLS=(n,n,n,n,n),Ix
```

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| IMb | These entries identify the initialize data cell program. The entries can optionally be omitted. |
| CELLS= | This entry identifies the parameter. |
| (n,n,n,n,n) | Each n is a decimal digit (from 0 to 9) designating a cell. From one (n) to five (n,n,n,n,n) cells can be specified in this parameter. |
| Ix IR<br>IA | IR indicates that previously flagged tracks are to retain their flags. If the Ix entry is omitted, IR is assumed.<br><br>IA indicates that the surfaces of all tracks are to be analyzed, ignoring any previous flags. |

## Label Control Set

Three types of label control cards, called a label control set, must be supplied:

- VTOC control card
- Volume label card
- END card.

VTOC CONTROL CARD

The VTOC control card provides the control information necessary for the creation of the VTOC. The card is punched in either standard (first example) or nonstandard (second example) format:

```
//bVTOCbSTANDARD
```

| Entry | Reason |
|---|---|
| //bVTOCb | Identifies this card as a VTOC control card. |
| STANDARD | Indicates that the VTOC is to be generated in the standard location. If processing other than a standard OPEN on the VTOC, the VTOC should be generated in a location other than the standard location. (Problem programs are not permitted to access subcell 0, strip 0, cylinder 0, track 0.) |

```
//bVTOCbSTRTADR=(ccccchhh),EXTENT=(n)
```

| Entry | Reason |
|---|---|
| //bVTOCb | Identifies this card as a VTOC control card. |
| STRTADR= | Indicates that this parameter contains the beginning address of the VTOC. |
| (c c c c h h h ) | Indicates the beginning address, where:<br><br>c =cell (0-9)<br>c c =subcell (00-19)<br>c =strip (0-9)<br>h =cylinder (0-4)<br>h h =track (00-19)<br><br>This parameter must be punched within the parentheses. |
| EXTENT= | Indicates this parameter contains the number of tracks in the VTOC. |
| (n) | Indicates the number of tracks (from 1 to 20) in the VTOC. This parameter must be punched within the parentheses. |

VOLUME LABEL CONTROL CARD

Contains information to create volume labels. A VOL1 control card must be supplied for each cell. Up to seven additional volume label control cards (VOL2 through VOL8) for user volume labels can be supplied for each cell. The VOL2 through VOL8 cards must be sequentially entered.

The VOLn label consists of the following fields:

col. 1- 3 Label identifier. Must contain VOL to indicate that it is a volume label.

col. 4 Volume label number. Indicates the relative position (1-8) of a volume label within a group of volume labels.

col. 5-10 Volume serial number. An identification code assigned to a volume when it enters an installation. Normally a numeric field from 000001 through 999999. Any or all of the six bytes may be alphameric.

col. 11 Volume security. Indicates the security status of the volume.

col. 12-21 Data file directory. The first five bytes contain the starting address of the VTOC, the last five bytes are blank. Used for DASD only.

col. 22-41 Reserved.

col. 42-51 Owner name and address code. Indicates a specific customer, installation, and/or system. Used by the VTOC Display program.

col. 52-80 Reserved.

For more information on the VOLn label, see the publication DOS Data Management Concepts, GC24-3427.

Columns 5 through 10 are used as the volume serial number. It is recommended that you always specify six characters in the volume label control card. If less than six characters are specified, however, the utility program pads the field to the right with blanks (X'40'). You must always ensure that all six characters are specified in job control extent information since the job control program pads an incomplete serial number to the left with zeros. Therefore, do not use // EXTENT if the serial number includes any blanks; otherwise, the rest of the information on the card is ignored and default values are assumed. Instead, use the VOL, DLAB, and XTENT statements whenever referencing a pack where blanks occur in the volume serial number. The // XTENT card enables you to specify the six-character volume serial number within quotes.

END CARD

The last label control set must be followed by an END card having the following format:

```
//bEND
```

## Control Statement Stream

A sample control statement input stream for running an initialize data cell program follows:

```
//bJOBbINIT
//bASSGNbSYS000,X'293'
//bEXECbINTM
//bUIMbCELLS=(3,5,7)
//bVTOCbSTRTADR=(3033303),EXTENT=(5)
VOL1222222
//bEND
//bVTOCbSTANDARD
VOL1333333
//bEND
//bVTOCbSTANDARD
VOL1444444
//bEND
/&
```

# Initialize Disk

The initialize disk program prepares one complete disk pack for use on the IBM 2311 Disk Storage Drive, the IBM 2314 Direct Access Storage Facility, or the IBM 2319 Disk Storage. The preparation of a pack consists of:

- VTOC label checking.
- Home address generation.
- Surface analysis and track descriptor record (R0) generation.
- Volume label creation.
- IPL and VTOC format creation.

This program can also be used to change the volume label(s) and VTOC address of a previously initialized disk pack (other than an emulator pack). Specifying IS in the input option parameter of the utility modifier statement will cause surface analysis, HA and R0 generation to be bypassed. A workpack used for OS can therefore be converted into a workpack suitable for DOS.

Note: If you want to use a 2314 pack as a 2311 pack, you must first initialize the 2314 as a 2314 and then initialize it as a 2311, because of end-of-cylinder conditions.

## VTOC Label Checking

Before a pack is initialized, it is checked to see if any labels present in the VTOC have expired. If the file has not expired, a message is printed. If you still wish to initialize the disk pack after receiving the message, you can ignore the label and continue to process; otherwise, the job is terminated.

## Home Address Generation

The home address is written by this program as follows: a flag (one byte), the cylinder number (two bytes), and the head number (two bytes). The flag byte is transmitted to the flag byte of each record on the track and indicates the condition of the track. Bits 0-5 are written as zero. Bit 6 is written as a zero if it is a good track or a one if it is a defective track. Bit 7 is written as a zero if it is not an alternate track and a one if it is an alternate track.

If the home address cannot be written on a track, the processing performed depends upon the device used:

1. IBM 2311: A message is printed to identify the home address.
2. IBM 2314/2319: The gap between the index marker and home address is extended by 705 bytes and an attempt is made to write the home address following the extended gap. If the home address cannot be written this time, a message is printed to identify the home address.

If no home address can be written on a pack, the job is cancelled.

## Surface Analysis and Track Descriptor Record (R0) Generation

Surface analysis is performed once on every track that is not already flagged defective. The utility modifier statement you supply may specify (1) surface analysis of all tracks including those already flagged defective, (2) tracks to be flagged defective without surface analysis, and (3) the number of times per track surface analysis is to be performed.

Surface analysis is performed first on the alternate tracks (cylinders 200-202) and then on the remaining tracks. Tracks found to be defective are flagged defective. The program flags a track defective by setting bit 6 of the flag byte in the home address to 1.

Alternate tracks are assigned to all defective tracks except defective alternate tracks (defective alternate tracks are not assigned). The program assigns an alternate track by writing (1) the alternate track's address in the track descriptor record of the defective track and (2) the defective track's address in the track descriptor record of the alternate track.

Processing is terminated when a defective track is found and no more alternate tracks are available. The defective tracks are logged to provide a record of the condition of each pack processed. When analysis has shown that a track is not defective, the track descriptor record (R0) is written. The track descriptor record is the first record

that follows the home address and is divided into two parts, count and data.

An eight-byte count field is written as follows: the cylinder number (two bytes), the head number (two bytes), the record number (one byte), the key length (one byte), and the data length (two bytes). The eight-byte data field is written with the cylinder number (two bytes), the head number (two bytes), the record number which is zero (one byte), the number of bytes remaining on the track (two bytes), and binary zeros (one byte).

If the track descriptor record cannot be written on a track, the processing performed depends upon the device used:

1. IBM 2311: A message is printed to identify the error.

2. IBM 2314/2319: The gap between the index marker and home address is extended by 705 bytes and an attempt is made to write the home address and track descriptor record again. If either cannot be written, the appropriate message is printed to identify the error.

The program continues analysis and track descriptor record generation to log any other defective tracks. A track descriptor record error results in job cancellation.

## Volume Label Creation

Through the use of a control card unique to this program, a volume label is created in the standard format (VOL1) for each pack processed. The volume label is written on cylinder 0, track 0, record 3 of the disk pack. Seven additional user volume labels (VOL2-VOL8) can be created, if desired, and placed in records 4-10.

## IPL and VTOC Creation

IPL FORMAT CREATION

This program formats two IPL records. These records are written on cylinder zero, track zero, records one and two. Record one is written with a 24-byte data field of binary zeros. Record two is written with a 144-byte data field of binary zeros.

VTOC FORMAT CREATION

The initialize disk program preformats the VTOC (volume table of contents). The location on the disk in which the VTOC is to be placed is indicated in a control card. A VTOC starting in the standard location and not extending to the end of cylinder 0 is therefore not possible. If a VTOC is required on cylinder 0 not extending the entire cylinder, it can be achieved by starting on track 1 (or track n).

The standard location of the VTOC is on cylinder zero immediately following the volume label(s) and extending to the end of the cylinder. However, the VTOC can appear on any cylinder (excluding alternate cylinders) but cannot exceed cylinder boundaries. A VTOC placed anywhere other than in the standard location can be any number of tracks desired on the cylinder. The first record begins in the first record of the first track, and the last record appears as the last record of the last track specified.

Each record of the VTOC contains a 44-byte key field and a 96-byte data field written as binary zeros. The first two records of the VTOC are reserved for two specific records. The first record is the format 4 label of the data set control block of the volume table of contents (VTOC-DSCB). The second record is for the space-management label (format 5).

The first four bytes of the key field of the space-management label are written as a hexadecimal five (05). The first byte of the data field is written in hexadecimal representation F5.

## Utility Modifier Statement

This statement, if used for this program, is read and diagnostics are performed. If this statement is omitted from the program, the following parameters are assumed:

```
//bUbIR,C1
```

Note: The default C1 is ignored if IS is specified.

The format and entries for the utility modifier statement for this program are:

```
//bUIDbIx,Cn,R=(ccccchhh)
```

Figure 25 shows detailed information of the entries in the utility modifier statement for the initialize disk program.

| Entry | Reason |
|---|---|
| //bU | These entries identify this as a utility modifier statement. |
| IDb | The initials of the program. These initials can be omitted. |

| PARAMETER | POSSIBLE FORMS | ENTRIES | EXPLANATION |
|---|---|---|---|
| Input Option Ix | IR<br>IA<br>IS | I | The first letter in this form identifies this as the input option parameter. |
| | | R | Previously flagged tracks are to retain their flags.<br>IR should be specified to avoid intermittent errors that may result when the pack is mounted on a different disk drive or when the circuitry is warming up.<br>If the Ix parameter is omitted, IR is assumed. |
| | | A | The surface of all tracks is to be analyzed, ignoring any previous flags. |
| | | S | Surface analysis, home address (HA) and R0 generation will be skipped. For initialized work packs this entry can be used to change the volume label(s) and VTOC address and/or to convert a workpack used for OS into a workpack suitable for DOS. When the program does not find a VOL1 label or a Format 4 label, the program will cancel the job. (The pack is thus assumed not to be initialized.)<br>The S entry in this input option parameter cannot be used for an emulator pack. |
| Count Option Cn | Cn | C | The first letter in this form identifies this as the count option parameter. |
| | | n | A decimal value of 1 to 255 should be entered, indicating the number of times surface analysis is to be performed. Enlarging this parameter causes a proportional increase in program run time. |
| Replace Option | R=(ccccchhh) | R= | This letter and symbol indicate this is the replace option parameter. This parameter is ignored if IS is specified. |
| | | (ccccchhh) | This entry indicates the physical location of a track that is to be unconditionally flagged as defective (cccc denotes the cylinder number and hhh denotes the head number). These numbers must be enclosed in parentheses. The replace option may be repeated any number of times up to the total number of alternate tracks available. The method of repeating this option is as follows:<br>R=(ccccchhh),R=(ccccchhh),R=(ccccchhh).<br>This option may be continued on additional cards by using the standard method for continuation cards, i.e., a character other than a blank is punched in column 72 and the continuation card is started in column 16. Note that each card must contain a full R-parameter. |

Figure 25. Initialize Disk Utility Modifier Statement

## Label Control Set

Three types of label control cards, called a label control set, must be supplied:

- VTOC control card
- Volume label control card
- END card.

VTOC CONTROL CARD

The VTOC control card provides the control information necessary for the creation of the VTOC. The card is punched in either standard (first example) or other than standard (second example) format:

//bVTOCbSTANDARD

| Entry | Reason |
|---|---|
| //bVTOCb | Identifies the card as a VTOC control card. |
| STANDARD | Indicates that the VTOC is generated in the standard location. If processing other than a standard OPEN on the VTOC, the VTOC should be generated in a location other than the standard location. (Problem programs are not permitted to access cylinder 0, track 0.) |

//bVTOCbSTRTADR=(ccccchhh),EXTENT=(n)

| Entry | Reason |
|---|---|
| //bVTOCb | Identifies the card as a VTOC control card. |
| STRTADR= | Indicates this parameter contains the beginning address of the VTOC. |
| (ccccchhh) | Indicates the beginning address (the cylinder and head numbers) and must be punched in decimal format within parentheses. |
| EXTENT= | Indicates this parameter contains the number of tracks in the VTOC. |
| (n) | Indicates the number of tracks in the VTOC and must be punched in decimal format within parentheses. |

VOLUME LABEL CONTROL CARD

The volume label control card contains the information for the creation of the volume label. A VOL1 volume label control card must be supplied. The VOL2 through VOL8 control cards for user volume labels can be supplied when user volume labels are desired. The VOL2 through VOL8 cards must be entered in sequence.

The VOLn label consists of the following fields:

| | | |
|---|---|---|
| col. | 1-3 | Label Identifier. (Must contain VOL to indicate that it is a volume label) |
| " | 4 | Volume Label Number. (Indicates the relative position (1-3) of a volume label within a group of volume labels) |
| " | 5-10 | Volume Serial Number. (An identification code assigned to a volume when it enters an installation. Normally a numeric field 000001 to 999999. However, any or all of the six bytes may be alphameric) |
| " | 11 | Volume Security. (Indicates security status of the volume) |
| " | 12-21 | Data File Directory. (The first five bytes contain the starting address (cchhr) of the VTOC, the last 5 bytes are blank. Used for DASD only) |
| " | 22-41 | Reserved. |
| " | 42-51 | Owner Name and Address Code. (Indicates a specific customer, installation and/or system. Used by the VTOC Display program.) |
| " | 52-80 | Reserved. |

For more information on the VOLn label see the SRL publication DOS Data Management Concepts.

The first six columns after "VOLn" are used as the volume serial number. It is recommended that you always specify six characters in the volume label control card. If less than six characters are specified, however, the utility program pads the field to the right with blanks (X'40'). You must always ensure that all six characters are specified in the job control extent information since the job control program pads an incomplete serial number to the left with zeros. Therefore, do not use // EXTENT if the serial number includes any blanks; otherwise, the rest of the information on the card is ignored and default values are assumed. Instead, use the VOL, DLAB, and XTENT statements whenever referencing a pack where blanks occur in the volume serial number. The // XTENT card enables you to specify the six-character volume serial number within quotes.

END CARD

An END card must follow the last card of the label control set:

```
//bEND
```

## Control Statement Stream

A sample control statement input stream for running an initialize disk program from the core image library follows; device and file descriptions are peculiar to the job being run.

```
//bJOBbEXAMPLE
//bASSGNbSYS000,X'191'
//bEXECbINTD
//bUIDbIR,C1,R=(0027003)
//bVTOCbSTANDARD
VOL1111111
//bEND
/&
```

Note: When initializing a pack to be used by the System/360 Model 30 or Model 40 Emulator (program numbers EU484, EU485), or by the System/370 Emulators of 1401, 1440, 1460, and 1410/7010 programs (program number EU490), you must include immediately before the EXEC card in the control statement stream an UPSI card of the following format:

```
//bUPSIb00000001
```

This card will allow cylinder 200 to be used for Model 30 or Model 40 Emulator data instead of being part of the alternate track area.

In order to perform standard label checking on tapes, standard volume labels must be present. The initialize tape program prepares up to eight volume labels, one dummy header label (HDR1 followed by binary zeros), and a tape mark for each tape submitted. No checking of the labels is performed after preparation. This program may be used to initialize either EBCDIC tapes with IBM standard volume labels or ASCII tapes with American National Standard Institute (ANSI) standard volume labels.

For EBCDIC tapes up to eight volume labels can be written per tape; for ASCII tapes, only one volume label can be written (ASCII tapes do not use VOL2-VOL8). The format of both standard volume labels is found in the DOS Data Management Concepts publication. There are two options for creating volume labels on tape:

1. A single control card may be used to provide the starting volume serial number, the owner name and address identification, and the protection code. This information will be written on the first tape supplied. The same volume label will be written on each succeeding tape supplied with the job, except that the volume serial number will be incremented by one.

2. Unique volume labels may be written on each tape by using a volume label image card. Up to eight volume label image cards may be supplied for EBCDIC tapes; only one volume label image card may be supplied for ASCII tapes. This card is the exact image of the 80 character label. This option should be used when the volume serial number field contains any alphameric values.

In addition to the volume label(s), this program will prepare a dummy header label immediately followed by a tape mark. As each tape is initialized, its volume label(s) and the tape unit number are printed on SYSLOG. This completes the initialization for one tape. The tape may then be either rewound or rewound and unloaded as specified by the user.

Tape units must be assigned in ascending sequence starting with SYS000. After the last unit has been assigned, the next sequential programmer logical unit should be unassigned (or assigned to a unit other than a tape drive) to insure proper termination. This avoids unintentional initialization.

If there are additional tape units on line (maximum is 16), the tape on the next higher unit (SYS001, SYS002, etc.) will be initialized. If there are no more additional units, one of the following actions will be taken:

1. The job is terminated if the rewind option was specified.

2. If the rewind option was not specified, the program waits until a new tape is mounted on the first output unit (SYS000) at which time tape initialization will begin a new cycle of the output units.

If the program reads the last card from the reader before completing a cycle of assigned units, the tape on the current unit is completed and the job is terminated.

## Job Control Cards

Upon initial program loading, symbolic names, channel addresses and tape characteristics for the initialize tape program must be defined through job control cards. These items once defined cannot be changed during the running of the program. If the required units for the program are not defined, the program will be terminated.

The following job control cards are used for system assignment:

| | |
|---|---|
| JOB card | Required. |
| ASSGN cards | Required as follows (if not permanently assigned): |
| SYSLOG | Must be assigned for output messages. |
| SYS000 | Must be assigned as first output tape unit. |
| SYS001-SYS015 | Optional. May be assigned as additional output tape units. |
| LOG card | Optional. |
| NOLOG card | Optional. |
| EXEC card | Required. |

## Utility Control Statements

Utility assignment for the initialize tape program is made by utility control

statements. The control statements and their associated positional parameters are as follows:

```
// INTT CARD,REWIND,A,SERIAL=(xxxxxx),
   P=(x),CODE=(xxxxxxxxxxxxxx)
```

Note: The parameters must be punched in the relative order above.

//b (Required.) Indicates control statement.

INTTb (Required.) Identifies an initialize tape control statement.

CARD (Optional.) This parameter indicates that the card image option is desired. If this option is specified, volume label image cards (only one card used for ASCII tapes) must be supplied. If this parameter is omitted, it is assumed that volume information for every tape initialized is to be taken from this control statement.

REWIND (Optional.) This parameter indicates that each tape is to be rewound at the completion of initialization. If this parameter is omitted, it is assumed that the tapes are to be rewound and unloaded at the completion of initialization.

A (Optional.) This parameter indicates that all tape labels are to be recorded in the ASCII mode and will observe the ANSI standards. If this parameter is omitted, EBCDIC mode and IBM standards will be assumed.

Note: If an optional parameter is not used, do not insert a comma for the omitted parameter. For example:

```
// INTT CARD,REWIND,SERIAL=(xxxxxx),
   CODE=(xxxxxxxxxxxxxx)
```

Card Image Option Not Specified

The following parameters are examined if the card image option is not specified:

SERIAL=(xxxxxx) (Required.) This is a six-character numeric field representing the starting value for volume serial number. This number is incremented by one for each additional tape initialized after the first.

P=(x) (Optional.) For EBCDIC tape, this parameter indicates that the volume security fields of all volumes are to be given the specified protection value. A value of 1 indicates protection. If this parameter is omitted, each volume will be given a value of zero, indicating no protection.

For ASCII tape, this parameter indicates that the accessibility fields of all volumes are to be given the specified protection value. A value of 1 indicates no accessibility. If this parameter is omitted, a space will be written, indicating unlimited accessibility.

CODE=(xxx...xxx) (Required.) For EBCDIC tape, this is a ten character alphameric value representing the owner's name and address code. For ASCII tape, this is a fourteen character value.

Card Image Option Specified

If the card image option is specified, the following control cards are required:

1. With EBCDIC tapes: from one to eight volume label image cards. These cards are an exact image of the 80-character label. They must be in ascending sequence (VOL1, VOL2, VOL3...etc.).

   With ASCII tapes: one volume label image card. This card is an exact image of the 80-character label. This card must contain VOL1 in the first four columns.

2. // END -- This card is used to separate the volume label image cards for different tapes. When this card is read, the program stops reading cards from the card reader until it finishes initializing the tape on the current output unit. If eight volume label image cards are supplied for a given EBCDIC tape, the // END card may be omitted.

The VOLn label consists of the following fields:

col. 1- 3 Label identifier. Must contain VOL to indicate that it is a volume label.

col. 4 Volume label number. Indicates the relative position (1-8) of a volume label within a group of volume labels.

col. 5-10 Volume serial number. An identification code assigned to a volume when it enters an installation. Normally a numeric field from 000001 through 999999. Any or all of the six bytes may be alphameric.

col. 11 Volume security. Indicates the security status of the volume.

col. 12-21 Data file directory. The first five bytes contain the starting address of the VTOC, the last five bytes are blank. Used for DASD only.

col. 22-41 Reserved.

col. 42-51 Owner name and address code. Indicates a specific customer, installation, and/or system. Used by the VTOC Display program.

col. 52-80 Reserved.

For more information on the VOLn label, see the publication DOS Data Management Concepts, GC24-3427.

Columns 5 through 10 are used for the volume serial number. It is recommended that you always specify six characters. If you specify less than six characters, the program pads the fields to the right with blanks (X'40'). Always ensure that all six characters are specified in the job control extent information, since the job control program pads an incomplete serial number to the left with zeros. Therefore, do not use // EXTENT if the serial number contains any blanks. Otherwise, the remainder of the information in the card is ignored and default values are assumed. Instead, use the VOL, DLAB, and EXTENT statements whenever blanks occur in the volume serial number. You can use the // EXTENT card to specify the six-character volume serial number within apostrophes.

## Control Statement Stream

Two sample control statement input streams for running an initialize tape program from the core image library follow; device and file descriptions are peculiar to the job being run. In each example SYSLOG is assumed to be permanently assigned. This example initializes an ASCII tape without using the card image option.

```
// JOB EXAMPLE
// ASSGN SYS000,X'181'
// ASSGN SYS001,UA
// EXEC INTT
// INTT REWIND,A,SERIAL=(000001),
   P=(1),CODE=(AB COMPANY NYC)
/&
```

This example initializes an ASCII tape using the card image option.

```
// JOB EXAMPLE
// ASSGN SYS000,X'181'
// ASSGN SYS001,X'182'
// ASSGN SYS002,UA

// EXEC INTT                                          (column 80)
// INTT CARD,A                                             ↓
VOL1000001              AB COMPANY NYC                      1
// END
VOL1000002              AB COMPANY NYC                      1
// END
/&
```

# Tape Compare

The tape compare program can be entered either in the core image or relocatable library. It compares two files from two or more tapes to ensure that the files are identical. The number of reels in each of the files need not be equal. Unlike the other utility programs described in this publication, the tape compare program requires 24K bytes of main storage.

The program does not perform tape positioning; therefore, the tapes are assumed to be positioned at the beginning of the run. If prepositioning of the tape is necessary before the compare operation, you may position the tapes by specifying that the tapes are unlabeled and by using the magnetic tape command (MTC) as found in the System Control and Service publication listed on the front cover of this publication.

Tapes containing fixed, variable, or undefined record lengths may be compared. When the tape compare program is initiated, it will normally run to completion regardless of the number of unequal compares that may occur. Unless a user exit has been specified for an unequal compare, any physical records that do not match will be written on SYSLST along with an index of the byte(s) that do not match, and the physical record number. No editing is performed on unprintable characters. If the exit has been specified, the tape compare program yields control through that exit.

Input areas are assigned from a common area of storage. The number of areas assigned to each file depends on the maximum size of the physical input records. If the space is available, two input areas are assigned; otherwise, one input area is assigned to each file.

If the tape files to be compared extend over more than one reel, the additional reels are also compared. If two tape drives are assigned for each file, the program can alternate between the two: for example, primary, alternate, primary, etc. In this case, tape reels are not rewound and unloaded. If only primary tape drives are assigned (and there are multiple reels per file), the operation waits for a new tape reel to be mounted on the primary tape drive.

The compare operation may be terminated at any time by pressing the external interrupt key. A compare operation for a new file can be initiated by supplying the correct control statement and following the restart procedures. This applies only when SYSIPT is assigned as a card reader. The program will automatically be terminated upon detection of the /* or /& control statement. The next job in SYSRDR will then be run.

## Restart Procedures

A restart procedure is available which allows you to control the program when the external interrupt feature is incorporated in the supervisor. The restart procedure is:

- Press the interrupt key, a message is printed, and the compare in process will be interrupted.
- Continue the current compare, start a new compare, or terminate the job by responding to the message with the appropriate character.

Any other information concerning the messages can be found in the appendix of this manual or the DOS Messages publication listed in the Preface.

## Label Processing

All volume labels are skipped without comparing. The first header and the first trailer file labels are checked to ensure that the file names are identical. Additional header, trailer, and user labels are bypassed. If the file names are not identical, both labels are printed.

When an end-of-volume (EOV) trailer label is sensed, the following action is taken:

- If the number of reels specified has not been processed, the compare continues on the next reel for the associated file.
- If the number of reels specified has been processed, the job will be terminated.

When an end-of-file (EOF) trailer label is sensed, the compare is terminated and you are given the option to restart or terminate the job.

## Nonstandard or Unlabeled Files

For nonstandard labels, if the first record from the tape is a tapemark, the tapemark is ignored. If a tape mark follows the nonstandard label, the reel count in the utility modifier statement must be a one; otherwise the data immediately following the label will not be compared. For every nonstandard label (with the following tapemark) detected for this file, the operator must supply another utility modifier statement with a reel count of one and restart the operation. Other tapemarks will be assumed to indicate an end-of-volume condition except when the reel count has been depleted, in which case the condition is assumed as an end-of-file condition. In any case, a compare operation may be restarted by supplying the correct control card and following the restart procedures.

For unlabeled files tapemarks will be assumed to indicate an end-of-volume condition except when the first record read from the tape is a tapemark, in which case the tapemark is ignored. An end-of-file condition is assumed when a tapemark has been detected and the reel count has been depleted. In any case, a compare operation may be restarted by supplying the correct card and following the restart procedures.

## Job Control Statements

Upon initial program loading job control statements define the symbolic names, channel addresses, and tape characteristics for the tape compare program. These items, once defined, cannot be changed during the running of the program. If the required units for the program are not defined, the program will be terminated.

The following job control statements are used for system assignment:

JOB card — Required.

ASSGN cards — Required as follows:

- SYSLOG — Must be assigned for diagnostic messages.
- SYSLST — Must be assigned for writing records that do not match (printer or tape).
- SYSIPT — Must be assigned for reading tape compare control statements (reader or tape).
- SYS004 — Must be assigned as the primary and alternate tape units for one of the tape files to be compared. This tape file will be referred to as file A.
- SYS005 — Must be assigned as the primary and alternate tape unit for the other file to be compared. This tape file will be referred to as file B.

## Linkage Editor

The program can be entered into either the relocatable or the core image library. If the program is in the relocatable library, the link edit phase must be performed.

The control statement stream used when executing a program that is resident in the relocatable library without a user exit routine is:

```
//bJOB
//bASSGN
//bOPTIONbLINK
bPHASEbTPCP,*,NOAUTO
bINCLUDEbIJWTCP
bINCLUDEbIJJCP0 (for TOS)
bINCLUDEbIJJCPD0 (for DOS)
bINCLUDEbIJWXIT
bINCLUDEbIJWTPCP
bENTRY
//bEXECbLNKEDT
//bEXEC
//bTPCP ...
.
.
/&
```

The control statement stream used when executing a program that is resident in the relocatable library with a user exit routine is:

```
//bJOB
//bASSGN
//bOPTIONbLINK
bPHASEbTPCP,*,NOAUTO
bINCLUDEbIJWTCP
bINCLUDEbIJJCP0 (for TOS)
bINCLUDEbIJJCPD0 (for DOS)
bINCLUDE (If the operand is omitted from
   this statement, the text of the user
   routine must be present on SYSIPT and
   followed by /* control statement. If
   the routine is in the relocatable
   library, it must have a user-assigned
   module name unique to the system as the
   operand.)
bINCLUDEbIJWTPCP
bENTRY
//bEXECbLNKEDT
//bEXEC
//bTPCP ...
.
.
/&
```

The control statement stream used when executing a program that is in the core image library is:

```
//bJOB
//bASSGN
//bEXECbTPCP
//bTPCP ...
.
.
/&
```

## Utility Control Statement

Utility assignment for the tape compare program is made by a utility control statement. Only one statement is used and it is read in by the mainline phase of the program. The control statement and its associated parameters are:

```
//bTPCPbRECSIZ=(m),LABELS,REELS=(n),ALTA,
   ALTB,EXIT
```

| Entry | Reason |
|---|---|
| //b (required) | Indicates a control statement. |
| TPCPb (required) | Identifies the tape compare control statement. |
| RECSIZ (required) | Identifies the record size parameter. |
| =(m) (required) | Maximum physical record size in bytes (that is, not logical record size). It must be enclosed in parentheses. This is needed for the assignment of input areas. If any physical input record exceeds this maximum, the excess is truncated and not compared. |
| LABELS (optional) | Indicates the tapes are labeled according to IBM System/360 standards. If this parameter is omitted, the tapes are assumed to be either unlabeled or not labeled according to IBM System/360 standards. In the latter case, the labels are treated as data. |
| REELS (optional) | Identifies reel count parameter to follow. |
| =(n) (optional) | Specifies the maximum number of reels per file to be compared. It must be enclosed in parentheses. If this parameter is omitted, n=1 will be assumed. n set to zero is an error. (Maximum value of n is 255.) If the tape file extends over more than one reel, this parameter must be used to cause the additional reels to be compared. |
| ALTA (optional) | Indicates an alternate unit for tape file A. If this entry is omitted, it is assumed that there is only a primary unit for tape file A. |
| ALTB (optional) | Indicates an alternate unit for tape file B. If this entry is omitted, it is assumed that there is only a primary unit for tape file B. |
| EXIT (optional) | Indicates that you wish the tape compare program to branch to a routine that you supply when an unequal compare is detected. If this entry is omitted, no branch will be made and unequal compare records are written. |

## User Exit Routine

If you supply an exit routine, the storage required for the routine is taken from the input area. If the exit routine is specified, the mainline phase branches through general register 2 to the location IJWXIT1 (defined as an entry point in the user's exit routine) when an unequal compare is sensed. Return to the tape compare program is through general register 14.

You have access to all physical and logical IOCS macro instructions to perform input/output, etc. The locations of the records that do not compare equally are supplied by general registers.

During user-exit routine processing program flow is:

1. Obtain the address of the file A description parameter list from register 0.
2. Obtain the address of the file B description parameter list from register 1.
3. Obtain the number of the mismatched record from register 10.

4. Perform user processing.

5. Return control to the tape compare program through register 14 (containing the return address).

File A Description Parameter List (Register 0)

The address of an eight byte parameter list is found in register 0. The first four bytes of the list contain the address of the file A input area; the second four bytes contain the length of the physical record.

File B Description Parameter List (Register 1)

The address of an eight byte parameter list is found in register 1. The first four bytes of the list contain the address of the file B input area; the second four bytes contain the length of the physical record.

## Control Statement Stream

Two sample control statement input streams for running from the disk resident core image and relocatable libraries follow; device and file descriptions are peculiar to the jobs being run.

Core Image Library

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'181'
//bASSGNbSYS005,X'182'
//bASSGNbSYS005,X'183',ALT
//bEXECbTPCP
//bTPCPbRECSIZ=(300),REELS=(2),ALTB
/&
```

Relocatable Library

```
//bJOBbEXAMPLE
//bASSGNbSYSLNK,X'180'
//bASSGNbSYS001,X'181'
//bOPTIONbLINK
bPHASEbTPCP,*,NOAUTO
bINCLUDEbIJWTCP
bINCLUDEbIJJCPD0
bINCLUDEbIJWXIT
bINCLUDEbIJWTPCP
bENTRY
//bEXECbLNKEDT
//bASSGNbSYS004,X'183'
//bASSGNbSYS005,X'184'
//bPAUSEbbOPERATORbPLACEbTAPEbAbCNbDRIVEb
   183bANDbTAPEbBbONbDRIVEb184
//bPAUSEbbRESTARTbJOBbBYbREPLYINGb2bTObEOFb
   MESSAGES
//bEXEC
//bTPCPbRECSIZ=(2000)
//bTPCPbRECSIZ=(2000)
//bTPCPbRECSIZ=(2000)
//bTPCPbRECSIZ=(2000)
/&
```

# VTOC Display

The VTOC program displays the labels contained in the volume table of contents of a disk pack on an IBM 2311 Disk Storage Drive, an IBM 2314 Direct Access Storage Facility, an IBM 2319 Disk Storage, or of a data cell on an IBM 2321 Data Cell Drive. The function of this display is to enable you to easily keep track of your files and their extents. The program provides for output to a printer, tape file, or disk pack.

The labels are identified by their location within the VTOC and their format types. All major fields are displayed along with appropriate heading lines.

The VOLn label consists of the following fields:

| | | |
|---|---|---|
| col. | 1-3 | Label Identifier. (Must contain VOL to indicate that it is a volume label) |
| " | 4 | Volume Label Number. (Indicates the relative position (1-3) of a volume label within a group of volume labels) |
| " | 5-10 | Volume Serial Number. (An identification code assigned to a volume when it enters an installation. Normally a numeric field 000001 to 999999. However, any or all of the six bytes may be alphameric) |
| " | 11 | Volume Security. (Indicates security status of the volume) |
| " | 12-21 | Data File Directory. (The first five bytes contain the starting address (cchhr) of the VTOC, the last 5 bytes are blank. Used for DASD only) |
| " | 22-41 | Reserved. |
| " | 42-51 | Owner Name and Address Code. (Indicates a specific customer, installation and/or system. Used by the VTOC Display program.) |
| " | 52-80 | Reserved. |

For more information on the VOLn label see the DOS Data Management Concepts publication listed in the Preface.

The first time a label of a data secured file is encountered, the message 8V96D FORMAT 1 LABEL OF DATA SECURED FILE is issued to the operator. The programmer instructs the operator to reply YES if the label information of all data secured files is to be included in the output listing. A data secured file is identified in the listing by the letters DSF directly following the format identification. Replying NC to the message causes the printing of label information for data secured files to be suppressed.

LINKAGE EDIT PROCEDURE

If the program is not already in the core image library, it must be link edited. Separate modules are supplied for link editing the VTOC display program with either a batch-only or a multiprogramming supervisor. The resultant multiprogramming supervisor program is self-relocating and may be executed in any partition.

With a batch-only supervisor, the following job stream should be used to link edit and catalog this program:

| | |
|---|---|
| //bJOBbBATCH | User-defined job name. |
| //bOPTIONbCATAL | Causes the program to be cataloged in the core image library. |
| bINCLUDEbIJWLVB | Used with a batch-only supervisor. This module is non-self-relocating. |
| //bLBLTYPbTAPE | Required if tape label processing is to be performed. |
| //bEXECbLNKEDT | Executes the linkage edit program. |
| /& | Defines the end of job. |

With a multiprogramming supervisor, the following job stream must be used to link edit, catalog, and execute this program. For self-relocating programs, storage is reserved for label processing at execution time, rather than at link-edit time. Therefore, any LBLTYP statement for this example must immediately precede the EXEC statement for the program.

| | |
|---|---|
| //bJOBbMULTI | |
| //bOPTIONbCATAL | |
| bINCLUDEbIJWLVM | Used with a multiprogramming supervisor. This module is self-relocating. |
| //bEXECbLNKEDT | |
| //bLBLTYPbTAPE | Required if tape label processing is to be performed. |
| //bEXECbLISTVTOC | |
| /& | |

CONTROL CARDS

No utility modifier card is needed for this program. A sample control statement input stream for running a VTOC display program from the core image library follows; device descriptions in the example are peculiar to the job being run. If there are any data secured files described in the VTOC, the operator will receive message 8V96D. The PAUSE card shown in this example indicates that such information should not be included in the printed output.

```
//bJOBbEXAMPLE
//bASSGNbSYS004,X'191'  (input on disk)
//bASSGNbSYS005,X'00E'  (output on printer)
//bPAUSE REPLY NO IF MSG 8V96D IS ISSUED
//bEXECbLISTVTOC
/&
```

Assignment of output allows you to assign SYS005 to a printer, tape, or disk. When SYS005 is a disk, DLBL and EXTENT descriptions peculiar to the device are required. When SYS005 is a tape, TLBL descriptions are required (standard labeled tape also requires LBLTYP statement if multiprogramming supervisor).

# Appendix A. Module Contents

File to File Utility Programs

Common Modules

IJJCP0 (for TOS); IJJCPD0 (for DOS)

Text for Phase 1, Part 2

IJWGEN

Text for Phase 2

IJWLAB

Text for Phase 5

Unique Program Modules

IJWxx

PHASE xxxx, *, NOAUTO

INCLUDE IJWxx1

INCLUDE IJJCP0 (for TOS);
INCLUDE IJJCPD0 (for DOS)

PHASE xxxx2, *, NOAUTO

INCLUDE IJWGEN

PHASE xxxx3, { IJWGENP2 / xxxx2 }, NOAUTO

(IJWGENP2 is for printer output)

INCLUDE IJWxx3

PHASE xxxx4, xxxx3, NOAUTO

INCLUDE IJWxx4

IJWxx1

Text for Phase 1, Part 1

IJWxx3

Text for Phase 3

IJWxx4

Text for Phase 4

Assign Alternate Track-Disk Modules

IJJCPD1N

Text for Phase 1, Part 1

IJWAD1

Text for Phase 1, Part 2

IJWAD2

Text for Phase 2

IJWAD3

Text for Phase 3

IJWAD4

Text for Phase 4

IJWAD5

Text for Phase 5

IJWAD

PHASE ATAD, *, NOAUTO

INCLUDE IJJCPD1N

INCLUDE IJWAD1

PHASE ATAD2, IJWAD1P1, NOAUTO

INCLUDE IJWAD2

PHASE ATAD3, IJWAD1P1, NOAUTO

INCLUDE IJWAD3

PHASE ATAD4, IJWAD1P1, NOAUTO

INCLUDE IJWAD4

PHASE ATAD5, IJWAD1P1, NOAUTO

INCLUDE IJWAD5

Note: xxxx represents the phase names and xx the module identification.

Assign Alternate Track - Data Cell Modules

```
IJJCPDIN

Text for Phase 1, Part 1
```

```
IJWAM1

Text for Phase 1, Part 2
```

```
IJWAM2

Text for Phase 2
```

```
IJWAM3

Text for Phase 3
```

```
IJWAM4

Text for Phase 4
```

```
IJWAM5

Text for Phase 5
```

```
IJWALTM


PHASE ATAM, *, NOAUTO
INCLUDE IJJCPDIN
INCLUDE IJWAM
PHASE ATAM2, IJWAM1P1, NOAUTO
INCLUDE IJWAM2
PHASE ATAM3, IJWAM1P1, NOAUTO
INCLUDE IJWAM3
PHASE ATAM4, IJWAM1P1, NOAUTO
INCLUDE IJWAM4
PHASE ATAM5, IJWAM1P1, NOAUTO
INCLUDE IJWAM5
```

Clear Disk and Clear Data Cell Module Contents

Common Modules

```
IJJCPD0
Text for Phase 1, Part 2
```

```
IJWCLD2
Text for Phase 2
```

```
IJWCLD3
Text for Phase 3
```

Clear Disk Modules

```
IJWCLD1
Text for Phase 1, Part 1
```

```
IJWCLD
PHASE  CLRDSK,*,NOAUTO
INCLUDE  IJWCLD1
INCLUDE IJJCPD0
PHASE  CLRD2,*,NOAUTO
INCLUDE  IJWCLD2
PHASE  CLRD3,CLRD2,NOAUTO
INCLUDE  IJWCLD3
```

Clear Data Cell Modules

```
IJWCLM1
Text for Phase 1, Part 1
```

```
IJWCLM
PHASE  CLDC,*,NOAUTO
INCLUDE  IJWCLM1
INCLUDE IJJCPD0
PHASE  CLDC2,*,NOAUTO
INCLUDE  IJWCLD2
PHASE  CLDC3,CLDC2,NOAUTO
INCLUDE  IJWCLD3
```

VTOC Display (Batched DOS Supervisor) and
VTOC Display (Multiprogramming Supervisor) Module Contents

Common Modules

```
IJWLV1
Text for Phase 1, Part 1
```

```
IJJCPD0N
Text for Phase 1, Part 2
```

```
IJWLVT
Transient Phase $$BLISTV
```

VTOC Display (Batched DOS Supervisor) Modules

```
IJWLVB
PHASE LISTVTOC, *, NOAUTO
INCLUDE IJWLV1
INCLUDE IJJCPD0N
INCLUDE IJWLVT
```

VTOC Display (Multiprogramming Supervisor) Modules

```
IJWLVM
PHASE LISTVTOC, +0, NOAUTO
INCLUDE IJWLV1
INCLUDE IJJCPD0N
INCLUDE IJWLVT
```

Common Module - Copy/Restore Programs

```
IJJCPD1N
Text for Phase 1, Part 1
```

Copy Disk to Card Modules

```
IJWKC1
Text for Phase 1, Part 2
```

```
IJWKC2
Text for Phase 2
```

```
IJWKC
PHASE CRDC, *, NOAUTO
INCLUDE IJJCPD1N
INCLUDE IJWKC1
PHASE CRDC2, IJWKC1OP, NOAUTO
INCLUDE IJWKC2
```

Copy Disk to Tape Modules

```
IJWKT1
Text for Phase 1, Part 2
```

```
IJWKT2
Text for Phase 2
```

```
IJWKT
PHASE CRDT, *, NOAUTO
INCLUDE IJJCPD1N
INCLUDE IJWKT1
PHASE CRDT2, IJWKT1OP, NOAUTO
INCLUDE IJWKT2
```

Copy Disk Modules

```
IJWRD1
Text for Phase 1, Part 2
```

```
IJWRD2
Text for Phase 2
```

```
IJWRD
PHASE CRDD, *, NOAUTO
INCLUDE IJJCPD1N
INCLUDE IJWRD1
PHASE CRDD2, IJWRD1OP, NOAUTO
INCLUDE IJWRD2
```

Restore Card to Disk Modules

```
IJWRC1
Text for Phase 1, Part 2
```

```
IJWRC
PHASE CRCD, *, NOAUTO
INCLUDE IJJCPD1N
INCLUDE IJWRC1
```

Restore Tape to Disk Modules

```
IJWRT1
Text for Phase 1, Part 2
```

```
IJWRT
PHASE CRTD, *, NOAUTO
INCLUDE IJJCPD1N
INCLUDE IJWRT1
```

Initialize Disk Modules

```
IJJCPD1N
Text for Phase 1, Part 1

IJWID1
Text for Phase 1, Part 2

IJWID2
Text for Phase 2

IJWID3
Text for Phase 3

IJWID4
Text for Phase 4

IJWID
PHASE INTD, *, NOAUTO
INCLUDE IJJCPD1N
INCLUDE IJWID1
PHASE INTD2, IJWID1P1, NOAUTO
INCLUDE IJWID2
PHASE INTD3, IJWID2P2, NOAUTO
INCLUDE IJWID3
PHASE INTD4, IJWID1P1, NOAUTO
INCLUDE IJWID4
```

Tape Compare Module Contents

```
IJJCP0 (for TOS); IJJCPD0 (for DOS)
Text of Phase 1, Part 2

IJWXIT
Text for Dummy User Routine (Phase 1, Part 3)

IJWTCP
Text for Phase 1, Part 1

IJWTCP2
Text for Phase 2

IJWTCP3
Text for Phase 3

IJWTPCP
PHASE TPCP2, *, NOAUTO
INCLUDE IJWTCP2
PHASE TPCP3, TPCP2, NOAUTO
INCLUDE IJWTCP3
```

Initialize Tape Module

```
IJWIT

PHASE INTT
INCLUDE IJWIT
```

Initialize Data Cell Modules

```
IJJCPDIN
Text for Phase 1, Part 1

IJWIM1
Text for Phase 1, Part 2

IJWIM2
Text for Phase 2

IJWIM3
Text for Phase 3

IJWIM4
Text for Phase 4

IJWIM

PHASE INTM, *, NOAUTO
INCLUDE IJJCPDIN
INCLUDE IJWIM1
PHASE INTM2, IJWID1P1, NOAUTO
INCLUDE IJWIM2
PHASE INTM3, IJWIM2P2, NOAUTO
INCLUDE IJWIM3
PHASE IJWIM4, IJWID1P1, NOAUTO
INCLUDE IJWIM4
```

# Appendix B. Printer Output

Note that the same data is used for both examples.

DATA LIST EXAMPLE

```
TAPE TO PRINT UTILITY
INPUT BLOCK LENGTH 00150
OUTPUT BLOCK LENGTH 00132
INPUT OPTION REWIND
OUTPUT OPTION PRINT CHARACTER
2 INPUT,2 OUTPUT AREAS ASSIGNED
RECORD FORMAT VARIABLE
TYPE LIST
STARTING RECORD NUMBER  00000001
8001D IS IT EOF
REPLY Y
NUMBER OF INPUT BLOCKS PROCESSED  000017
NUMBER OF OUTPUT BLOCKS PROCESSED 000020
END OF JOB
```

Job descriptive (logging) messages as it appeared on symbolic device SYSLST.

```
THIS IS A SAMPLE OF THE HEADING INFORMATION LINE THAT MAY BE USED IF THE USER DESIRES.  DATA LIST WITH CHARACTER
 I  5ID000165I000165I00000000165000001651DDDD
```

DATA DISPLAY EXAMPLE

```
TAPE TO PRINT UTILITY
INPUT BLOCK LENGTH 00150
OUTPUT BLOCK LENGTH 00120
INPUT OPTION REWIND
OUTPUT OPTION PRINT CHARACTER
2 INPUT,2 OUTPUT AREAS ASSIGNED
RECORD FORMAT VARIABLE
TYPE DATA DISPLAY
STARTING RECORD NUMBER  00000001
8001D IS IT EOF
REPLY Y
NUMBER OF INPUT BLOCKS PROCESSED  000017
NUMBER OF OUTPUT BLOCKS PROCESSED 000017
END OF JOB
```

Job descriptive (logging) messages as it appeared on symbolic device SYSLST.

```
 THIS IS A SAMPLE OF THE HEADING INFORMATION LINE THAT MAY BE USED IF THE USER DESIRES.  DATA DISPLAY WITH CHARACTER
  BL SZ  BL NO  RC NO....5....1....5....2....5....3....5....4....5....5....5....6....5....7....5....8....5....9....5.....
       77     1    1  (   I  51D000165100016510000000016500000165IDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
       75     2    2  .   G  8AD275228A275228A000002752280027522 8ADDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
       84     3    3  M   &  1AD685901A685901A000006859010868590 1ADDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
       64     4    4      '  2ED636222E636222E000006362228863622 2EDDDDDDDDDDDDDDDDDDDDDD
       98     5    5  S      9ED342279E342279E000003422795234227 9EDDDDDDDDDDDDDDDDDDD
```

# Appendix C. File—to—File Program Messages

The following are file-to-file program messages that appear on the device (either a printer or a tape unit) assigned to SYSLST. The messages are divided into three groups:

- Diagnostic messages
- Processing messages
- Information messages

A job is terminated when a diagnostic message is received; the operator is informed of this condition on the SYSLOG device. When informational and processing messages are received, processing continues.

<table>
<tr><th colspan="3">RESPECTIVE ORDER OF DIAGNOSTIC MESSAGES FOR THE FILE-TO-FILE PROGRAMS</th></tr>
<tr><td colspan="3">Note: Whenever xxx precedes a message, it indicates in which field definition the error occurred, e.g. cards 1 and 2 each have 5 field definitions: for a format error in the third definition,xxx would be printed as 003; for a format error on the fifth definition of card 2, xxx would be printed as a cumulative 010.</td></tr>
<tr><th>MESSAGE</th><th>REASON</th><th>ACTION</th></tr>
<tr><td>END CARD MISSING</td><td>No END statement supplied (// END),or noncontrol statement read before END.</td><td>Job is cancelled.</td></tr>
<tr><td>x INVALID FORMAT. UTILITY MODIFIER CARD</td><td>Format specifications for utility modifier statement were not followed or all required parameters were not supplied as follows:<br><br>x: Decoded message<br><br>A: Error in input format specifications (A parameter).<br>B: Error in output format specifications (B parameter).<br>E: Error in device type specification (E parameter).<br>a. Invalid format.<br>b. Not consistent with program type in utility modifier card.<br>F: Error in record format specifications (F parameter).<br>I: Invalid input option (I parameter).<br>J: Invalid type of job (J parameter).<br>M: Missing required parameter (F,A,B parameters must be present).<br>N: Invalid type of program (U identifier // U not found, or xx representing the program type is not valid).<br>O: Invalid output option (O parameter).<br>P: Invalid page number option (P parameter).<br>Q: Error in sequence checking specifications (Q parameter).<br>R: Error in starting record specifications (R parameter).<br>S: Invalid spacing option (S parameter).<br>T: Type of job parameter missing (T parameter).<br>U: Undefined parameters (parameter identifier not valid).</td><td></td></tr>
</table>

| MESSAGE | REASON | ACTION |
|---|---|---|
| FIELD SELECT CARD MISSING | Field select was indicated on utility modifier statement, but no field select statement was supplied. | Job is cancelled. |
| xxx INVALID FORMAT FIELD SELECT CARD | Format specifications for field select statement were not followed. (000 indicates no fields for field select were indicated but CV was present.) | |
| FIELD SELECT CARD NOT EXPECTED | Field select was not indicated on utility modifier statement, but field select statement was supplied. | |
| INVALID CONTROL CARD | A control statement (with //b in the first 3 columns) was read which was not a utility modifier, field select, print header, or END statement. | |
| INVALID INPUT DEVICE AT SYS004 | The device assigned to SYS004 is not valid for this program. | |
| INVALID OUTPUT DEVICE AT SYS005 | The device assigned to SYS005 is not valid for this program. | |
| UNDEFINED FORMAT CAN ONLY DISPLAY | Data display is the only mode that can be indicated for undefined records in printer output programs. | |
| xxx CANNOT FIELD SELECT INTO lst 4 CHARACTERS | The indicated field cannot be selected into the record length field of a variable-length record. | |
| INVALID OUTPUT DEVICE AT SYS006 | The device assigned to SYS006 is not valid for this program. | |
| UNDEFINED FORMAT CAN ONLY COPY | Copy is the only format that can be indicated for undefined records in non-printer program. | |
| INCORRECT PROGRAM | Utility modifier statement punched with the wrong program initials, such as DT for a disk to card program. | Job is cancelled. Note that all succeeding messages may not have a valid meaning. |
| x INVALID FORMAT UTIL MOD CARD | x: Utility modifier statement error | Job is cancelled. |
| | A: For non DASD input, a key field was used.<br>B: For nonprinter output, a printer B format was used; for non DASD output, a key field was used.<br>K: For non-DASD input or output, a key field was used. | |
| FIXED LENGTH RECORD FORMAT REQUIRED | Card input or card output was not fixed length. | |

| MESSAGE | REASON | | | ACTION |
|---|---|---|---|---|
| INVALID JOB FOR THIS PROGRAM | Program | Valid Types | Invalid Types | Job is cancelled. |
| | Undefined records | | | |
| | a. TP,DP and MP | *D | C,B,BF,F,L, LF,R,RF | |
| | b. DD,DM,DT, MD,MM,MT, TD,TM, and TT | C | B,BF,D,F,L, LF,R,RF | |
| | Fixed-length records without key fields | | | |
| | a. CP | B,BF,C,*D, F,L,LF | R,RF | |
| | b. MP,TP, and DP | *D,L,LF | B,BF,C,F,R, RF | |
| | c. CD,CT,DC, DD,DM,DT, MD,MM,MT, TC,TD,TM and TT | C,F,R, RF | B,BF,D,L,LF | |
| | Fixed-length records with key fields | | | |
| | a. CD,DC | F | B,BF,C,D,L, LF,R,RF | |
| | b. DT,MT,TM, and TD | F,RF | B,BF,C,D,L, LF,R, | |
| | c. DD,MM,DM, and MD | C,F | B,BF,D,L,LF, R,RF | |
| | d. DP and MP | *D,L,LF | B,BF,C,F,R,RF | |
| | Variable-length records without key fields | | | |
| | a. MP,TP, and DP | *D,L,LF | B,BF,C,F,R RF | |
| | b. DD,DM,DT, MD,MM,MT, TD and TT | C,F,R, RF | B,BF,D,L,LF | |
| | * If first character forms control is specified (S parameter), data display is invalid. | | | |

| MESSAGE | REASON | ACTION |
|---|---|---|
| INVALID INPUT RECORD LENGTH | a. Card input. Record length was greater than 80 (EBCDIC) or 160 (binary).<br>b. Tape input. Record length was greater than 8192.<br>c. DASD input without key. Block length was not a multiple of the record length.<br>d. DASD record length exceeds 3625 for 2311, 7294 for 2314 and 2319, or 2000 for data cell. | Job is cancelled. |
| NONSTANDARD LABEL INVALID INPUT | DASD programs do not allow nonstandard labels. | |
| NONSTANDARD LABEL INVALID OUTPUT | | |
| INVALID INPUT OPTION | Option is incorrect for the program. No option for DASD input. | |
| INVALID OUTPUT OPTION | Option is incorrect for the program. | |
| INVALID CARD SEQUENCE | Card programs. The length parameter specified is over 10 characters or the starting position plus the length exceeds 80 characters. | |
| I/O AREA CANNOT BE ASSIGNED | Not enough main storage to assign the specified input/output areas. | |
| FIELD SELECT MUST BE SPECIFIED | When the output record length differs from the input record length, field select must be used. For printer programs (list function) the input record length cannot exceed the size of the print line. For DASD programs with key fields (except DASD to printer or DASD to DASD) field select must be specified. | |
| xxx INVALID UNPACK OUTPUT LENGTH | The parameter values specified are invalid. | |
| xxx INVALID PACK OUTPUT LENGTH | | |
| xxx RECORD CAPACITY EXCEEDED BY PACK | The xxxth field select parameter specifies a field not entirely contained within the input or output record. | |
| xxx RECORD CAPACITY EXCEEDED BY UNPK | | |
| xxx RECORD CAPACITY EXCEEDED BY FS | | |
| xxx RECORD CAPACITY EXCEEDED BY HEX | | |

| MESSAGE | REASON | ACTION |
|---|---|---|
| xxx FIELD SELECT PARAMETER FOR NONEXISTENT KEY | A key field was specified in the field select statement, but no key was indicated in the utility modified statement. | Job is cancelled. |
| INVALID OUTPUT RECORD LENGTH | a. Card output. Record length was greater than 80 (EBCDIC) or 160 (binary).<br>b. Tape output. Record length was greater than 8192.<br>c. Printer output. Record length was greater than 144.<br>d. DASD output. The output record length is greater than 3625 for 2311, 7294 for 2314 and 2319, or 2000 for data cell. | |
| INVALID INPUT KEY LENGTH | For a DASD input the key length is greater than 255. | |
| INVALID OUTPUT KEY LENGTH | For a DASD output the key length is greater than 255. | |
| INVALID INPUT BLOCK LENGTH | a. Card input. Block and record length are not equal.<br>b. Tape input. For fixed length record processing the input block length was not a multiple of the record length; otherwise, the block length was not 4 greater than the fixed portion.<br>c. DASD input. Input block length is greater than 3625 for 2311, 7294 for 2314 and 2319, or 2000 for data cell. | |
| INVALID OUTPUT BLOCK LENGTH | a. Block length is not a multiple of the record length.<br>b. For DASD the output block length is greater than 3625 for 2311, 7294 for 2314 and 2319, or 2000 for data cell.<br>c. For the copy function the block lengths must be equal. | |
| INVALID INPUT DATA LENGTH | DASD input programs with key require data length plus key length to be less than or equal to 3605 for 2311, 7249 for 2314 and 2319, or 1984 for data cell. | |
| INVALID OUTPUT DATA LENGTH | DASD output programs with key require data length plus key length to be less than or equal to 3605 for 2311, 7249 for 2314 and 2319, or 1984 for data cell. | |
| xxx FS INPUT LENGTH EQUALS ZERO | Input field length has been specified as zero. | |
| xxx PACK INPUT LENGTH EQUALS ZERO | | |
| xxx UNPK INPUT LENGTH EQUALS ZERO | | |

| MESSAGE | REASON | ACTION |
|---|---|---|
| xxx HEX INPUT LENGTH EQUALS ZERO | | Job is cancelled. |
| xxx CANNOT PROCESS HEX PARAMETER | Hexadecimal indicator valid only for print output programs. | |
| xxx CANNOT PROCESS PACK PARAMETER | Cannot pack a field for print output programs. | |
| USER ROUTINE NOT PRESENT | User label checking is specified on the UPSI statement, but a user label routine is not present. | |

RESPECTIVE ORDER OF FILE-TO-FILE PROCESSING MESSAGES

| Messages (on SYSLST) | Format | Function | Primary Condition | Associated Conditions | Processing |
|---|---|---|---|---|---|
| BLOCK NO. xxxxxx, INPUT AREA OVERFLOW | F, V, or U | Copy | Input block length is longer than that specified in the utility modifier statement. | None | The specified input block size is copied and the remainder is truncated. If the records are variable length, the count field is not corrected. |
| BLOCK NO. xxxxxx, INPUT AREA UNDERFLOW | F | Copy | Input block length is shorter than that specified in the utility modifier statement. | | Only the actual block size is copied (no padding). |
| BLOCK NO. xxxxxx, INPUT AREA UNDERFLOW | F | R, F, RF, L, or LF | | The actual block size is a multiple of the specified record size but less than the specified block size. | Processing is performed as specified for the short block. This message is not issued if the starting record number in the record-skipping parameter has not been encountered. |
| BLOCK NO. xxxxxx, INPUT AREA UNDERFLOW BLOCK NO. xxxxxx, RCD. NO. xx RECORD AND REMAINDER OF BLOCK DROPPED | F | R, F, RF, L, or LF | | The last logical record of the input block is less than the specified record size. | Processing is normal up to the short record. The record is dropped and processing continues. This message is not issued if the starting record number in the record-skipping parameter has not been encountered. The short record is counted as one. |

| MESSAGE | | | REASON | | ACTION |
|---|---|---|---|---|---|
| Message (on SYSLST) | Format | Function | Primary Condition | Associated Conditions | Processing |
| BLOCK NO. xxxxxx, INPUT AREA OVERFLOW | V | R, F, RF, L, or LF | Input block length is longer than that specified in the utility modifier statement. | The last position of the specified block is the last position of a logical record. | The overflow records from the input block are truncated. This message is issued even if the first record to be processed has not been reached. The truncated records are not counted. |
| BLOCK NO. xxxxxx, INPUT AREA OVERFLOW BLOCK NO. xxxxxx, RCD. NO. xx RECORD AND REMAINDER OF BLOCK DROPPED | V | R, F, RF, L, LF | | The last logical record in the specified block size is not complete within the block. | The input block (and the last logical record) are truncated. The truncated record is dropped. The second message is not issued if the starting record number in the record skipping parameter has not been encountered. The dropped part of the block is counted as one. |
| BLOCK NO. xxxxxx, RCD. NO. xx RECORD AND REMAINDER OF BLOCK DROPPED | V | R, F, RF, L, or LF | An input logical record contains an invalid length field. A record length field is invalid if it is less than 5 or is not equal to the number of bytes read. | | Processing of the current block cannot proceed and the block is dropped. This message is issued even if the record-skipping parameter number has not been reached. The part of the block is counted as one. |
| BLOCK NO. xxxxxx, RCD. NO. xx, SHORT VARIABLE LENGTH RECORD DROPPED | V | F, RF, or LF | The length of a logical input record is less than that specified as the fixed portion of the variable-length records. | | The record is dropped and processing continues with the next record, if present. This message is not issued if the record-skipping parameter has not been encountered. The dropped record is counted as one. |

| MESSAGE | | | REASON | ACTION | |
|---|---|---|---|---|---|
| Messages (on SYSLST) | Format | Function | Primary Condition | Associated Conditions | Processing |
| BLOCK NO. xxxxxx, OUTPUT AREA OVERFLOW | V | R, F, RF, L, or LF | A generated output record exceeds the block size specified in the utility modifier statement. | | The generated block is truncated. The block count and record count are corrected and the block written out. |
| BLOCK NO. xxxxxx, KEY LENGTH IS xxx | F or V | C,R,F, RF,L, or LF | The key length for this block is invalid, or it differs from the key length specified in the utility modifier statement. | a. For undefined records, the message should not occur.<br><br>b. For fixed-length records with no key fields specified, or variable length records, only the data portions are processed.<br><br>c. For fixed-length records with key fields specified, the actual and specified key length differ. Both key and data fields are processed as specified (i.e., if the actual key is less than that specified, the difference is made up with data bytes; if greater, the excess is treated as data bytes.) | Processing continues, with the output record formatted as specified in the utility modifier statement (if valid specification). |

| RESPECTIVE ORDER OF FILE-TO-FILE INFORMATIONAL MESSAGES | |
|---|---|
| Control parameter diagnostics are followed by logging messages in this order. | |
| MESSAGE | ACTION |
| CARD TO DISK<br>CARD TO PRINTER/PUNCH<br>CARD TO TAPE<br>DATA CELL TO DATA CELL<br>DATA CELL TO DISK<br>DATA CELL TO PRINTER<br>DATA CELL TO TAPE<br>DISK TO CARD<br>DISK TO DATA CELL<br>DISK TO DISK<br>DISK TO PRINTER<br>DISK TO TAPE<br>TAPE TO CARD<br>TAPE TO DATA CELL<br>TAPE TO DISK<br>TAPE TO PRINT<br>TAPE TO TAPE<br>UTILITY | Identifies the particular utility program. The program continues processing. |
| INPUT { FIXED PORTION xxxx / KEY LENGTH xxxx / DATA LENGTH xxxxx / RECORD LENGTH xxxx / BLOCK LENGTH xxxx } | Processing continues. (x represents a digit.) |
| OUTPUT { FIXED PORTION xxxx / KEY LENGTH xxxx / DATA LENGTH xxxxx / RECORD LENGTH xxxx / BLOCK LENGTH xxxx } | |
| INPUT { CARD BCD / CARD BINARY / NO REWIND, UNLOAD / REWIND / REWIND, UNLOAD } | |
| OUTPUT OPTION { BCD, CHARACTER / CARD BCD / CARD BINARY / DISK WRITE CHECK / NO DISK WRITE CHECK / PRINT CHARACTER / PRINT HEX / NO REWIND, UNLOAD {WRITE TAPE MARK} / REWIND {WRITE TAPE MARK} / REWIND, UNLOAD {WRITE TAPE MARK} } | |
| { x INPUT, x OUTPUT / x INPUT/OUTPUT } AREAS ASSIGNED | |
| RECORD FORMAT { FIXED / VARIABLE / UNDEFINED } | |

| MESSAGE | ACTION |
|---|---|
| TYPE { COPY / DATA DISPLAY / FIELD SELECT / LIST / LIST, FIELD SELECT / PRINT AND PUNCH / PRINT, PUNCH, FIELD SELECT / REBLOCK / REBLOCK, FIELD SELECT } | Processing continues. (x represents a digit.) |
| STARTING SEQUENCE COLUMN xx | |
| SEQUENCE LENGTH xx | |
| STARTING RECORD NUMBER xxxxxxxx | |
| INPUT DEVICE TYPE xxxx | These messages are printed for DASD devices. xxxx indicates 2311, 2314 * or 2321.<br>* 2314 is also printed when a 2319 is used. |
| OUTPUT DEVICE TYPE xxxx | |
| REPLY x | This message is printed to indicate the reply given to a diagnostic printed on SYSLOG. The action taken is indicated by the letter x. Processing continues. |
| 1ST CHARACTER FORMS CONTROL TYPE { A / B / C / D } | Processing continues. |
| xx ERRORS FOUND IN CONTROL CARDS | |
| CARD SEQUENCE ERROR, CURRENT SEQ xxxxxxxxxx<br>LAST SEQ xxxxxxxxxx | |
| END OF DATA | END OF DATA will not be printed for first-character forms control. |
| FILE MARK WRITTEN IN<br>XT. NO. B1 C1 C2 H1 H2 R<br>xxx xxx xxx xxx xxx xxx xxx | For DASD output programs the decimal value of the EXTENT sequence number and the address of the file mark (written at the end of the file) are logged. The headings represent bin (B1), subcell (C1), strip (C2), cylinder (H1), track (H2), and record (R) numbers for data cell. For disk, they represent cylinder (C2), track (H2), and record (R) numbers. |
| NUMBER OF { INPUT / OUTPUT } BLOCKS PROCESSED xxxxxx | Processing continues. |
| SPECIFIED STARTING RECORD NO. LARGER THAN TOTAL NO. OF LOGICAL INPUT RECORDS | |
| END OF JOB | |

# Appendix D. Tape Compare, Copy/Restore, and Initialize Tape Program Messages

The following are tape compare diagnostic messages that appear on the device assigned to SYSLST. This device can be either a printer or a tape unit.

| MESSAGE | REASON | ACTION |
|---|---|---|
| INVALID INPUT DEVICE AT SYS004 | The device assigned to SYS004 is not valid for this program. | Job is cancelled. |
| INVALID OUTPUT DEVICE AT SYS005 | The device assigned to SYS005 is not valid for this program. | |
| REC. NO. xxxxxxx, CAUSEy<br><br>xxx100 characters of file A dataxxx<br><br>xxx100 characters of file B dataxxx<br><br>---------index line------*----- | The record number identifies the data records. The reason (CAUSE) the data is printed is identified by the character y which may be:<br><br>4-the records do not compare.<br><br>6-one of both records are greater than the maximum specified.<br><br>7-the records are not of an equal length.<br><br>The priority of the messages is 7, 6, then 4.<br><br>The index line refers to each character compared. A (-) indicates the characters were equal; an (*) indicates an unequal compare. | |

The following are copy/restore informational messages that appear on the device assigned to SYSLST.

| MESSAGE | ACTION |
|---|---|
| FILE COPY REQUESTED VOLUME | Processing continues. |
| RECORD SIZE EQUALS nnnn (nnnn replaced by decimal record size) | |
| IPL OPTION REQUESTED | |
| INPUT DEVICE is a 23xx (xx is 11 or 14)* | |

*2314 is also printed when a 2319 is used

The following are the diagnostic messages fcr the initialize tape program that appear on the device assigned to SYSLOG. See page 169 for more initialize tape messages.

| MESSAGE | REASON | ACTION |
|---|---|---|
| INCORRECT CARD | Volume-image card is missing or first three characters are not VOL. | Job is cancelled. |
| CARD READ WAS NOT INTT | Utility control statement does not start with // INTT. | |
| NO TAPES ASSIGNED | No tape unit was assigned to SYS000 (required as the first assignment). | |
| SERIAL NOT OR INCORRECTLY SPECIFIED | SERIAL parameter is missing or incorrectly specified in the utility control statement. | |
| CODE NOT OR INCORRECTLY SPECIFIED | CODE parameter is missing or incorrectly specified in the utility control statement. | |
| END OF JOB | End-of-job for the utility program. | Processing continues. |
| ABOVE VOL CARDS DID NOT PROCESS | An error was detected in volume-image cards. | Job is cancelled. |
| SEQUENCE ERROR DETECTED | Volume-image cards are not in the proper sequence. | |
| PASS nn OF TAPES ASSIGNED | Initialization cycle nn is started. | Processing continues. |
| SYSnnn | Indicates the number of the logical unit on which a tape reel has been initialized. (This message is followed by the printing of the label written onto the tape.) | |

# Appendix E. Clear Data Cell and Clear Disk Program Messages

The following are informational or diagnostic messages that appear on the device assigned to SYSLST. This device can be either a printer or a tape unit.

| MESSAGE | REASON | ACTION |
|---|---|---|
| CLEAR DATA CELL UTILITY<br>CLEAR DISK UTILITY | The name of the program is logged for identification. | Processing continues. |
| UTILITY CONTROL CARDS | This heading message immediately precedes the logging of the control cards. | |
| INVALID CARD | Valid utility control cards begin with //bU; //bEND; or with ./bU; ./bEND. | Job is cancelled. |
| INVALID PARAMETER | Valid parameters begin with B, C, X, O, and E. None of these parameter identifiers may be repeated with the control card, nor may C and X appear together. | |
| INVALID FORMAT | The format of at least one of the above parameters is incorrect; e.g., the key and data lengths must be specified as B=(K=1 to 3 digits, D=1 to 4 digits). | |
| INVALID KEY LENGTH | The key length must be ≥ 0 and ≤ 255. | |
| INVALID DATA LENGTH | The data length must be greater than 0. If a key length specification is greater than 0, the key length plus the data length must be ≤1984 for data cell, ≤3605 for 2311, or ≤ 7249 for 2314 and 2319. If a key length specification is equal to 0, the data length must be ≤ 2000 for data cell, ≤ 3625 for 2311, or ≤ 7294 for 2314 and 2319. | |
| INVALID OUTPUT PARAMETER | Valid output parameter values are OY or ON. | |
| INVALID DEVICE TYPE PARAMETER | Valid device type parameters are 2311, 2314*, and 2321.<br>*If a 2319 is used, the valid parameter is 2314. | |
| I/O AREA NOT ENOUGH FOR SPECIFIED RECORD SIZE | The block size specified in the utility modifier statement exceeds the main storage available. | |
| SPECIFIED PARAMETERS | This heading message identifies the specified utility modifier statement parameters. | Processing continues. |
| ASSUMED PARAMETERS | This heading message identifies the assumed utility modifier statement parameters. | |

| MESSAGE | REASON | ACTION |
|---|---|---|
| CYL xxx TRACK xxx<br>IS IN ERROR | For the referenced track, a no record found or a track overflow condition has been posted. Records on this track may be invalid. | Processing Continues. |
| NO END CARD | Either no END card was supplied (//bEND), or a noncontrol statement was read before END. | Job is cancelled. |

Informational messages are logged in this order.

```
KEY LENGTH - xxx
DATA LENGTH - xxxx
FILL CHARACTER - {X'xx'}
                 {C'x' }
OUTPUT PARAMETER - x
DEVICE TYPE - xxxx
RECORDS/TRACK - xx
   EXTENT     BB          LOWER LIMIT                    UPPER LIMIT
SEQ. NO.                 C1   C2   H1   H2          C1   C2   H1   H2
   xxx       xxx       xxx  xxx  xxx  xxx         xxx  xxx  xxx  xxx
   xxx       xxx       xxx  xxx  xxx  xxx         xxx  xxx  xxx  xxx
    .         .         .    .    .    .           .    .    .    .
    .         .         .    .    .    .           .    .    .    .
    .         .         .    .    .    .           .    .    .    .

END OF JOB
```

# Appendix F. Operator Communication Messages

## File-to-File Messages

8001D IS_IT_EOF

Cause: Tape input is specified as unlabeled. A tape mark was encountered on the input file tape.

System Action: The system waits for an operator response.

Programmer Action: Not applicable.

Operator Action: Type Y if end of file, (the file does not reside on the next volume), or

Type N if end of volume. (Processing should continue on next volume.)

Default System Action: End of file assumed.

8002A PUNCH_CHECK

Cause: A punch check occurred on the card read punch (2520 or 2540).

This is probably a hardware error.

System Action: The system waits for an operator response.

Programmer Action: Not applicable.

Operator Action: Run cut cards in punch, discard last three or four cards (for the 2520, one punched and two blank cards; for the 2540, two punched and two blank cards). Ready the punch and type any character to continue processing.

If the problem recurs, issue the ROD command, execute EREP and retain the listing to complete your problem determination action.

Default System Action: After the punch is restarted, processing continues. The card in error and the following cards are repunched at the point where the punch check occurred.

## Tape Compare Messages

8003A ALTA_OR_ALTB_PARAMETER_SPECIFIED TWICE

Cause: The ALTA or ALTB parameter in the tape compare utility control statement has been specified twice.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: If the problem recurs, do the following to complete your problem determination action:

1. Take a main storage dump at once and retain the listing.
2. Have the system log, job stream, and printer output available.

Operator Action: Repunch the utility control statement (// TPCP...) specifying ALTA or ALTB only once, and enter the corrected card on SYSIPT. Type 2 on the console typewriter to continue processing.

If there is one alternate tape unit for both file A and file B, the utility control statement should include both ALTA and ALTB, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8004I // TPCP_RECSIZ=_(nnnnn)

Cause: The physical record size (in bytes) specified in the utility control statement is printed on the console typewriter.

This message is issued for information only.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8005A // TPCP RECSIZ= (FORMAT IS INCORRECT)

Cause: Control statement format is invalid.

This is probably a user error.

Programmer Action: Not applicable.

Operator Action: Repunch the utility control statement (// TPCP...) using the correct format and enter the corrected card on SYSIPT. Type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate job.

If the problem recurs, do the following to complete your problem determination action:

1. Take a main storage dump and retain the listing.
2. Have the job stream, printer output, and system log available.

Default System Action: The job is cancelled.

8006A RECORD SIZE OR REEL COUNT PARAMETER MISSING

Cause: The value for the RECSIZ or REELS parameter on the utility control statement is missing.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: If the problem recurs, do the following to complete your problem determination action:

1. Take a main storage dump at once and retain the listing.
2. Have the job stream, system log, and printer output available.

Operator Action: Repunch the utility control statement (// TPCP...) with the correct record size or reel count and enter the corrected card on SYSIPT. Type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8007A INVALID RECORD SIZE OR REEL COUNT PARAMETER

Cause: Record size is greater than five digits, or reel count exceeds 255.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: If the problem recurs, do the following to complete your problem determination action:

1. Take a main storage dump at once and retain the listing.
2. Have the system log, printer output, and job stream available.

Operator Action: Repunch the utility control statement (// TPCP...) supplying the correct record size or reel count, and enter the corrected card on SYSIPT. Type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8008A LEADING ZERO IN RECORD SIZE OR REEL COUNT PARAMETER

Cause: A leading zero is invalid in a control statement parameter.

This is probably a user error.

2. Have the console log, printer output, and the job stream available.

Operator Action: Repunch the utility control statement (// TPCP...) reducing the specified record size to the capacity of the I/O area, and enter the corrected card on SYSIPT. Type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8012A USER EXIT SPECIFIED BUT NONE SUPPLIED

Cause: The EXIT parameter has been specified in the utility control statement but no user exit routine has been supplied in the card deck.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Supply the user exit routine or remove the EXIT parameter from the utility control statement.

If the problem recurs, do the following to complete your problem determination action:

1. Take a main storage dump and retain the listing.
2. Have the job stream, printer output and system log available.

Operator Action: Repunch the utility control statement (// TPCP...) omitting the EXIT parameter, and enter it on SYSIPT. Then type 2 on the console typewriter to continue processing.

Type any character other than 2 to terminate job.

Default System Action: The job is cancelled.

8013A INVALID TPMK DETECTED ON FILE n

Cause: An unexpected tapemark encountered on file A or B:

- Labeled files were specified and a tapemark preceded the label, or
- Two tapemarks preceded either the first data record or the trailer label.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Take a tape to printer dump of the label area to check labels.

If the problem recurs, do the following to complete your problem determination action:

1. Take a main storage dump at once and retain the listing.
2. Have the job stream, printer output, system log, and label area dump listing available.

Operator Action: Mount the correct tapes, or

Correct the assignment of the logical unit, or

Repunch the utility control statement to reflect unlabeled tapes and enter the corrected statement on SYSIPT. Type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8014A VOLUME LABEL MISSING ON FILE n

Cause: Standard labels are indicated in the utility control statement but no label was found on file A or B.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: If the tape should contain standard labels, take a tape to printer dump of the label area and verify that a VOL1 label is the first label on the tape.

If the problem recurs, do the following to complete your problem determination action:

1. Retain the label area listing.
2. Take a main storage dump at once and retain the listing.
3. Have the system log, the printer output, and the job stream available.

Operator Action: Mount the correct tape, or

Correct the assignment of the logical unit, or

If the tape is unlabeled, repunch the utility control statement omitting the LABELS parameter, and enter the corrected statement on SYSIPT. Type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8015A HEADER LABEL MISSING ON FILE n

Cause: Standard labels are indicated in the utility control statement but no header label was found on file A or B.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: If the tape should contain standard labels, take a tape to printer dump of the label area and verify that a header label is on the tape.

If the problem recurs, do the following to complete your problem determination action:

1. Retain the label area listing.
2. Take a main storage dump at once and retain the listing.
3. Have the system log, the printer output, and the job stream available.

Operator Action: Mount the correct tape, or

Correct the assignment of the logical unit, or

If the tape is unlabeled, repunch the utility control statement omitting the LABELS parameter, and enter the corrected statement on SYSIPT. Type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8016A TRAILER LABEL MISSING ON FILE n

Cause: Standard labels are indicated in the utility control statement but no trailer label was found on file A or B.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: If the tape should contain standard labels, take a tape to printer dump of the label area and verify that EOV or EOF appears on the tape.

If the problem recurs, do the following to complete your problem determination action:

1. Retain the label area listing.
2. Take a main storage dump at once and retain the listing.
3. Have the system log, the printer output, and the job stream available.

Operator Action: Mount the correct tape, or

Correct the assignment of the logical unit, or

If the tape is unlabeled, repunch the utility control statement omitting the LABELS parameter, and enter the corrected statement on SYSIPT. Type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The jcb is cancelled.

8017D EOF ON UNLABELED FILES

Cause: A tapemark was detected on an unlabeled file and the reel count is depleted.

The job step is completed.

System Action: The system waits for an operator response.

Programmer Action: Not applicable.

Operator Action: Supply control statement on SYSIPT and type 2 to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The jcb is cancelled.

8018D EOF ON FILE A AND NOT ON B

Cause: File A is shorter than file B.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Verify that the correct tapes were supplied with the job.

If the problem recurs do the following to complete your problem determination action:

1. Have listings of the tape files or records in question available.
2. Have the job stream, printer output, and system log available.

Operator Action: Mount the correct tapes, reenter the utility control statement on SYSIPT, and type a 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8019D EOF ON FILE B AND NOT ON A

Cause: File B is shorter than file A.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Verify that the correct tapes were supplied with the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have listings of the tape files or records in question available.
2. Have the job stream, system log, and printer output available.

Operator Action: Mount the correct tapes, reenter the utility control statement on SYSIPT, and type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8020A CHANGE REEL ON PRIMARY A

Cause: An alternate reel was not assigned to primary A and EOV was detected.

System Action: The system waits for an operator response.

Programmer Action: Not applicable.

Operator Action: Change the reel and type any character to continue processing.

8021I SWITCHING TO ALTERNATE A

Cause: Primary reel is completed and processing continues with alternate reel.

This messages is for information only.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8022A CHANGE REEL ON PRIMARY B

Cause: An alternate reel was not assigned to primary B and EOV was detected.

System Action: The system waits for an operator response.

Programmer Action: Not applicable.

Operator Action: Change the reel and type any character to continue processing.

8023I SWITCHING TO ALTERNATE B

Cause: Primary reel is completed and processing continues with alternate reel.

This message is for information only.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8024D REEL COUNT DEPLETED

Cause: The reel count is depleted on a labeled file and no EOF trailer label was sensed.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: If the problem recurs, have the job stream, program output, system log, and printer output available to complete your problem determination action.

Operator Action: Repunch the utility control statement (// TPCP...) supplying the correct REELS parameter, and enter the corrected card on SYSIPT. Type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8025A RESTART WAS REQUESTED

Cause: The interrupt key was pressed during execution.

System Action: The system waits for an operator response.

Programmer Action: Not applicable.

Operator Action: Type a blank to continue processing, or

Supply new control statement on SYSIPT and type 2 to restart, or

Type any character other than blank or 2 to terminate the job.

Default System Action: The job is cancelled.

8026D EOF ON LABELED FILES

Cause: An end-of-file trailer label has been detected on both files.

System Action: The system waits for an operator response.

Programmer Action: Not applicable.

Operator Action: Mount new tapes, supply a new utility control statement to SYSIPT, and type 2 on the console typewriter to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

8027A CONTROL CARD MISSING

Cause: TPCP control statement was omitted.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: If the problem recurs, have the job stream, printer output, and the system log available to complete your problem determination action.

Operator Action: Supply IPCP control statement on SYSIPT and type 2 to continue processing, or

Type any character other than 2 to terminate the job.

Default System Action: The job is cancelled.

## Copy/Restore Disk or Data Cell Messages

8050I NOT A STD R0 RECORD

Cause: A nonstandard R0 record was encountered on disk input file, indicating that the disk was not properly initialized. A pack with a nonstandard R0 cannot be copied.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: Instruct the operator to mount the correct disk pack and resubmit the job, or

Initialize the defective disk pack, recreate the file, and resubmit the job.

If the problem recurs, have the job stream, program listing, log sheet, and printer output available to complete your problem determination action.

Operator Action: Check that the proper disk pack is mounted and that the assignments are correct. If corrections are necessary, rerun the job.

8051I NOT A STD R0 RECORD

Cause: A nonstandard R0 record was encountered on disk output file, indicating that the disk was not properly initialized. A pack with a nonstandard R0 cannot be restored.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: Check that the proper mounting and assignment instructions were given, or

Initialize the output disk pack. Resubmit the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, program listing, log sheet, and printer output available.
2. Execute a stand-alone dump at the time of the failure and save the output.

Operator Action: Check that the proper disk pack is mounted and that the assignments are correct. If corrections are necessary, rerun the job.

8052D RECORD GREATER THAN I/O AREA

Cause: The size of the record read is greater than the size of the available I/O area.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: If the record can be truncated, resubmit the job and instruct the operator to reply 2 to this message, or

Resubmit the job with a request for a larger partition allocation.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Type 2 to truncate record and continue processing, or

Type any character other than 2 to cancel the job. Allocate a larger partition and rerun the job.

Default System Action: The job is cancelled.

8053I I/O AREA INSUFFICIENT

Cause: Insufficient I/O area available for the indicated average record size.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Resubmit the job with a request for a larger partition.

If the problem recurs, have the job stream, log sheet, and printer output available, to complete your problem determination action.

Operator Action: Allocate a larger partition and rerun the job.

8054I NO_VOL1_LABEL

Cause: No VOL1 label was found while searching for the VTOC address. The file has been destroyed or does not exist on that volume.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Assign the correct volume and resubmit the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute LISTVTOC for the subject volume and have the output available.
3. Execute a stand-alone dump at the time of the failure.

Operator Action: Mount the correct pack and resubmit the job.

8055I SYS005_NOT_ASSIGNED

Cause: A tape was not assigned to SYS005 as an output unit.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Resubmit the job with the correct assignment for SYS005.

Operator Action: Assign SYS005 to the output tape and rerun the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the LISTIO command and have the output available.

8056I IPL_SPECIFIED_AND_NOT_FOUND

Cause: No IPL records were found when the option was requested for the copy file.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Omit the IPL parameter in the utility modifier statement or assign the input to the correct copy file.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute the disk to printer utility to obtain a listing of cylinder 0, track 0.

Operator Action: Check that the correct copy file has been mounted. If correction is necessary, rerun the job.

8057I TAPE_RECORD_GREATER_THAN_MAX_I/O_AREA

Cause: The tape record being restored is greater than the maximum I/O area available. The program that created the tape was probably run in a larger partition.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Request a greater storage allocation for the problem program and resubmit the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute a tape to printer utility for the record in question and have the output available.

Operator Action: Allocate a larger partition and rerun the job.

8058A INPUT IS OUT OF SEQUENCE

Cause: The card input is out of sequence, or

The tape reel is out of sequence.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Check the input card deck for correct sequence or the tape reels for proper mounting sequence.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute a tape to printer utility for the record in question and have the output available.
3. Execute a stand-alone dump at the time of the failure.

Operator Action: Correct card sequence and type 2 to continue processing, or

Mount a new tape and type 2 to continue processing. If an alternate tape is assigned, the new tape must be placed on the next assigned tape drive, or

Type any character other than 2 to cancel the job.

Default System Action: The job is cancelled.

8059A READER OUT OF INPUT

Cause: The card reader is out of cards.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Check the card deck to insure that all necessary cards were supplied. Make any necessary corrections and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Supply additional card input and type 2 to continue processing, or

Type any character other than 2 to cancel the job.

Default System Action: The job is cancelled.

8060I SYS004 NOT ASSIGNED

Cause: A tape was not assigned to SYS004 as an input unit, or

A card reader was not assigned to SYS004, or

A disk was not assigned to SYS004 for a copy volume function.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Issue the LISTIO command to check the assignments, assign SYS004 to the input unit, and rerun the job.

If the problem recurs, have the job stream, log sheet, printer output and LISTIO output available to complete your problem determination action.

8061I CONTROL RECORD NOT FOUND

Cause: The first data record read was not a control record. The control record is written by the copy program and contains parameters unique to this file. This message can occur if a tape is mounted out of sequence, the wrong tape is mounted, or the wrong pack is mounted.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the input and resubmit the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute a tape to printer utility for the record or file in question and have the output available.

Operator Action: Check that the correct pack or tape is mounted. If correction is necessary, rerun the job.

8062I PARTITION TOO SMALL

Cause: The size of the restore partition is less than that required by the copy program.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Allocate sufficient main storage for this application and rerun the job.

8063I SYS006 NOT ASSIGNED

Cause: A card punch was not assigned to SYS006.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Execute LISTIO to check the assignments and then assign SYS006 to a card punch and rerun the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

8064I ERRORS IN CONTROL CARD

Cause: Errors were detected in the utility modifier card.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the errors in the utility modifier card and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8065I RESTORE EXTENTS NOT EQUAL TO COPY

Cause: The EXTENT limits for the disk output file do not include the EXTENT limits for the disk input file. EXTENTS are all used for output file, but records still remain on input file.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Execute LISTVTOC and check the EXTENTS allocated. Increase or correct the restore EXTENTS so that they are equal to or greater than the copy EXTENTS.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Have the LISTVTOC output available.

Operator Action: Not applicable.

8066I END OF COPY

Cause: Normal end-of-job indication.

System Action: Normal system processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8067I END OF RESTORE

Cause: Normal end-of-job indications.

System Action: Normal system processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8068I CHECK POINT BEING TAKEN FOLLOWING CARD NO. xxxxxx

Cause: A checkpoint record is being written on SYS003 following the processing of the referenced card.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8070I INCORRECT CONTROL IDENTIFIER

Cause: The control card is not properly identified.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Check the // EXEC card for the correct utility name and check for keypunch errors in the identifier parameter of the utility modifier statement.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

8071I INCORRECT [I, E, M, O] OPTION

Cause: An invalid entry was made for the indicated parameter.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Check for keypunch errors in the utility modifier card. If correction is necessary, rerun the job.

If the problem recurs, have the job stream, program listing, log sheet, and printer output available to complete your problem determination action.

8072I INCORRECT FORMAT

Cause: Incorrect parameter separation was used, or

The parameter was punched incorrectly.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Check for the indicated errors in the utility modifier statement and resubmit the job.

If the problem recurs have the job stream and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8073I INVALID LEADING ZERO IN SIZE PARAMETER

Cause: Preceding zeros in A=(a) are invalid.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Omit the leading zeros in the A=(a) parameter and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator action: Not applicable.

8074I INCORRECT CHARACTER IN SIZE PARAMETER

Cause: Only numeric values are acceptable in the A=(a) parameter.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Check that only numeric values are used in the A=(a) parameter. Make the necessary corrections and resubmit the job.

If the problem recurs, have the job stream, program listing, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8075I A PARAMETER TOO LARGE

Cause: The A=(a) parameter exceeds the maximum value for the device.

| Device | Record Size (in bytes) | Problem Program Area |
|---|---|---|
| 2311 | up to 3625 | 10K |
| 2314/2319 | up to 6400 | 10K |
| 2311 | up to 3625 | 12K |
| 2314/2319 | up to 7294 | 12K |
| 2314/2319 | up to 7294 | 18K |
| 2321 | up to 2000 | 10K |

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Reduce the A=(a) parameter to the maximum permitted and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8076I INCORRECT PARAMETER

Cause: A character, other than the first, of a parameter in the utility modifier statement is in error.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the parameter in error and resubmit the job.

If the problem recurs, have the job stream, program listing, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8077I DUPLICATE [A, I, M, T, E, O] PARAMETER

Cause: A second entry in the card began with one of the letters of an entry already processed.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the utility modifier statement and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8079I SIZE PARAMETER MISSING or [A, T] PARAMETER MISSING

Cause: The (a) within the required A=(a) was not specified, or

The required F or V was not specified with the Tt parameter.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the utility control statement by providing the size parameter [(a)], or the A or T parameter. Resubmit the job.

If the problem recurs, have the job stream, program listing, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8081I IPL OPTION INVALID FOR COPY VOLUME FUNCTION

Cause: The IPL parameter was supplied for the copy volume function. The parameter is treated as invalid since it may only be used for a copy file function.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Omit the IPL option in the utility modifier card and resubmit the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, program listing, log sheet and printer output available.
2. Execute a stand-alone dump at the time of the failure and save the output.

Operator Action: Not applicable.

8082I UTILITY MODIFIER CARD MISSING

Cause: A utility modifier card was not supplied.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Supply a utility modifier statement and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8083I DUPLICATE ENTRIES IN CELLS PARAMETER

Cause: The same cell number was specified more than once in the CELLS parameter. Duplicate entries are not permitted.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Supply the correct cell number and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8084I EXCESSIVE NUMBER OF CELLS PARAMETER ENTRIES

Cause: More than five cell numbers were specified in the CELLS parameter.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the CELLS parameter so that it specifies five or fewer cells and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8085I CELLS MUST BE PROCESSED IN THIS ORDER . . . .

Cause: Indicates the order in which cells were copied.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8086I EMULATOR PACK

Cause: Pack was initialized for a System/360 Model 30/40 or System/370 emulator.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: The applicable 1400 utility program must be used to copy emulator packs. If a restore is being done, a pack initialized for System/360 or System/370 must be used.

Operator Action: Check that the correct input and output files are mounted. If correction is necessary, rerun the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

## Initialize Disk and Data Cell Utility Messages

The numeric values in the comments of these messages, represented by small x's, are in hexadecimal notation.

8101I SYS000 NOT ASSIGNED TO A 2311 OR 2314

Cause: A 2311, 2314, or 2319 is not assigned to SYS000. 2314 is printed for both the 2314 and the 2319.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Assign SYS000 to a 2311, 2314, or 2319 and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Assign SYS000 to a 2311, 2314, or 2319 and rerun the job.

8102I UTILITY MODIFIER CARD

Cause: This is an information only message.

System Action: The utility modifier control card parameters are listed following this message and processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8103I INVALID CARD

Cause: The utility modifier statement was improperly identified.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the utility modifier statement and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8104I INVALID FORMAT

Cause: A required parameter on the utility modifier card is either missing or out of sequence.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the utility modifier statement and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8105I INVALID PARAMETER

Cause: The parameter value in the utility modifier card is incorrect.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the utility modifier parameter in error and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8106I INVALID USE OF S ENTRY IN INPUT OPTION

Cause: The S entry in the input option parameter (that is, IS) indicates that surface analysis and R0 generation are to be skipped. The error was caused by either:

1. The Initialize Disk program cannot find a VOL1 or Format 4 label and assumes that the pack has not been initialized, or
2. IS was specified while initializing an emulator pack.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Change the input option parameter or supply the correct disk pack and rerun the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute LISTVTOC for the subject volume and save the output.

Operator Action: Insure that the correct disk pack has been mounted and that the assignment is correct. If correction is necessary, rerun the job.

8107I CYLxx, TRKxx, IS A DEFECTIVE ALTERNATE TRACK

Cause: The indicated alternate track is defective and will not be assigned.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8108I CYLxx, TRKxx, IS DEFECTIVE AND AN ALTERNATE IS ASSIGNED

Cause: The main area of the indicated track is defective and an alternate is assigned.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8109I CYLxx, TRKxx, IS DEFECTIVE AND NO ALTERNATE IS AVAILABLE

Cause: The indicated track is defective and no more alternates are available.

This is possibly a hardware error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: If the message 8108I has occurred several times prior to this message, and:

- The indicated track on all these messages is the same, the disk drive is probably defective. Move the disk pack to another drive and rerun the program.
- The indicated tracks are random, the pack is probably defective.

If the problem persists, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.

8110I CYLxx, TRKxx, HA or REC0 IS IN ERROR

Cause: The portion of the track where HA or Record 0 is written is defective.

This is possibly a hardware error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Mount a new disk pack and rerun the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.

8111A VTOC CARD MISSING

Cause: VTOC card is missing cr incorrect.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Insert or correct the VTOC card and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Correct the card, place it in the reader, and ready the reader. Type 2 to continue processing, or

Type any character other than 2 tc cancel the job.

Default System Action: The jcb is cancelled.

8112A VTOC ADDRESS INVALID

Cause: The VTOC start address is invalid, or

An EXTENT parameter is invalid or missing.

System Action: The system waits for an operator response.

Programmer Action: Correct the VTOC control card, or correct or supply an EXTENT parameter and resubmit the job.

If the problem recurs, have the jcb stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Correct the card, place it in the reader, and ready the reader. Type 2 to continue processing, or

Type any character other than 2 to cancel the job.

Default System Action: The job is cancelled.

8113A VTOC OVERFLOWS CYLINDER

Cause: Assigned VTOC area overflows the cylinder.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Correct the job stream so that the VTOC area is one cylinder or less and resubmit the job.

If the problem recurs, have the jcb stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Correct the card, place it in the reader, and ready the reader. Type 2 to continue processing, or

Type any character other than 2 to cancel the job.

Default System Action: The job is cancelled.

8114A VOL CARD MISSING

Cause: VOL1 card is missing, incorrect, or out of sequence.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Correct the placement, omission, or contents of the VOL1 card and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Correct the card, place it in the card reader, and ready the reader. Type 2 to continue processing, or

Type any character other than 2 to cancel the job.

Default System Action: The jcb is cancelled.

8115A VOL1 SERIAL FIELD

Cause: VOL1 card has all blanks in the volume serial field.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Correct the VOL1 card by completing cclumns 5-10 and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Correct the card, place it in the card reader, and ready the reader. Type 2 to continue processing, or

Type any character other than 2 to cancel the job.

Default System Action: The jcb is cancelled.

8116A VTOC OR END CARD ERROR

Cause: A VTOC or END card is incorrect or an END card is missing.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Correct the VTOC or END card, or insert the missing END card. Resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Correct the card, place it in the card reader, and ready the reader. Type 2 to ccntinue processing, or

Type any character other than 2 to cancel the job.

Default System Action: The job is cancelled.

8117A PARAMETER DELIMITER

Cause: A comma is missing after a parameter.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Insert the missing comma and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Correct the card, place it in the card reader, and ready the reader. Type 2 to continue processing, or

Type any character other than 2 to cancel the job.

Default System Action: The job is cancelled.

8118D UNEXPIRED FILE

Cause: An unexpired file was detected on SYS000.

This is probably a user error.

System Action: The system waits for an operator response.

Programmer Action: Check that the information on the unexpired file is not to be retained or substitute different extents. Resubmit the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute LISTVTOC for the subject file and save the output.

Operator Action: Type 2 to continue the job and to delete this or any other unexpired file, or

Type any character other than 2 to cancel the job.

Default System Action: The job is cancelled.

8120I END OF INIT. [DISK, DATA CELL]

Cause: Normal end of initialize disk or data cell program.

System Action: The job is finished.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8121I UNRECOVERABLE DISK ERROR

Cause: An unrecoverable disk error occurred while perfcrming surface analysis.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Mount another disk pack and rerun the jcb.

If the problem persists, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.

8122I LABEL CONTROL SET

Cause: The label control cards are printed after this message.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8123I EMULATOR PACK, STANDARD VTOC ASSUMED

Cause: A nonstandard VTOC has been specified for an emulator pack, although it requires a standard VTOC.

System Action: Processing ccntinues. Standard VTOC is built for emulator pack.

Programmer Action: Not applicable.

Operator Action: Not applicable.

## Alternate Track Assign Utility Messages

The numeric values in the comments of these messages, represented by small x's, are in hexadecimal notation.

8201I SYS000 NOT A VALID DISK DRIVE

Cause: A disk was not assigned to SYS000.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Assign SYS000 to a disk drive and rerun the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

8203I INVALID CARD

Cause: The utility modifier statement is improperly identified.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the utility modifier statement and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Check the // EXEC statement against the utility modifier card for the correct program. Correct the utility modifier statement and rerun the job.

8205I INVALID FORMAT

Cause: A parameter is missing or out of sequence.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Correct the utility modifier card and rerun the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

8206I INVALID PARAMETER

Cause: A parameter in the utility modifier statement was incorrectly specified.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the parameter value in the utility modifier statement and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8207I UTILITY MODIFIER CARD

Cause: The control card parameters are listed following this message.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8210I FORMAT 4 LABEL MISSING

Cause: The VOL1 label contains the address of the VTOC, but no Format 4 label can be found at that address.

This is possibly a hardware error.

System Action: The job is cancelled.

Programmer Action: Recreate the file on another disk pack or data cell, or reinitialize this disk pack or data cell and recreate the file.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.
3. Execute LISTVTOC for the subject volume and save the output.

Operator Action: Check that the proper disk pack or data cell has been mounted. If not, mount the proper volume and rerun the job.

8211I VOLUME LABEL MISSING

Cause: The volume label is always record 3 of cylinder 0, track 0 for a disk, or record 3 of subcell 0, strip 0, cylinder 0, track 0 for a data cell. The volume label cannot be found at this location.

System Action: The job is cancelled.

Programmer Action: Reinitialize the disk pack and recreate the file.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output avilable.
2. Issue the ROD command, execute EREP, and save the output.
3. Execute LISTVTOC for the subject volume and save the output.

Operator Action: Check that the proper disk pack or data cell has been mounted. If not, mount the proper volume and rerun the job.

8212I DATA CHECK IN LABEL

Cause: A data check occurred in the count field while reading a label.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: If the problem recurs, the file should be recreated on another disk pack or data cell and the subject volume should be reinitialized.

Operator Action: Rerun the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.

8213D FORMAT 4 LABEL ERROR

Cause: An error occurred while reading a Format 4 label.

This is probably a hardware error.

System Action: The system waits for an operator response.

Programmer Action: If the error recurs, recreate the file on another disk pack or data cell and reinitialize the subject volume.

Operator Action: Type a 2 to continue processing, or

Type any character other than 2 to cancel the job, or

Mount the pack on another disk drive and rerun the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute a stand-alone dump at the time of the failure and save the output.
3. Issue the ROD command, execute EREP, and save the output.
4. Execute LISTVTOC for the subject volume and save the output.

Default System Action: The job is cancelled.

8214D VOLUME LABEL ERROR

Cause: An error occurred while reading a volume label.

This is probably a hardware error.

System Action: The system waits for an operator response.

Programmer Action: If the problem recurs, recreate the file on another disk pack or data cell and reinitialize the subject volume.

Operator Action: Type a 2 to continue processing, or

Type any character other than 2 to cancel the job, or

Mount the pack on another drive and rerun the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.

2. Execute a stand-alone dump at the time of the failure and save the output.

3. Issue the ROD command, execute EREP, and save the output.

Default System Action: The job is cancelled.

8215I ALT CYLS FULL

Cause: No more alternate tracks are available for assignment.

System Action: The job is cancelled.

Programmer Action: The file should be recreated on ancther disk pack, or

This pack should be initialized and a complete surface analysis performed before rebuilding the file.

If the problem recurs, do the following to complete your prcblem determination action:

1. Have the job stream, log sheet, and printer output available.

2. Issue the ROD command, execute EREP, and have the output available.

Operator Action: Not applicable.

8216I CYLxx, TRKxx REC0 IN ERROR

Cause: The portion of the track on which record 0 is written is defective.

System Action: The job is cancelled.

Programmer Action: The file should be recreated on ancther disk, or

The subject pack should be reinitialized and the file should be rebuilt.

If the problem recurs, do the following to complete your prcblem determination action:

1. Have the job stream, log sheet, and printer output available.

2. Issue the ROD command, execute EREP, and save the output.

3. Execute the disk to printer utility for the record in question.

Operator Action: Not applicable.

8220I ccchhhhrrkkdddd

Cause: If there are no errors, the eight byte count field is printed in hexadecimal as each record is transferred.

c=cylinder
h=head
r=record
k=key
d=data

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8221I ALT TRK ASSIGNED NOT ACCESSIBLE

Cause: The HA and R0 area designated is defective. The alternate track is not accessible for the valid data.

This is possibly a hardware error.

System Action: The job is cancelled.

Programmer Action: The file should be created on another disk pack or data cell, or

This disk pack or data cell should be initialized and the file recreated.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.

2. Issue the ROD command, execute EREP, and save the output.

Operator Action: Not applicable.

8222I HA AND R0 ARE DEFECTIVE

Cause: The HA and R0 areas are defective. An alternate track was not previously assigned, and therefore all the records will be printed on SYSLST regardless of print option.

System Action: Processing continues.

Programmer Action: Recreate the file on another disk pack or data cell, or initialize this disk pack or data cell.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.

Operator Action: Not applicable.

8223I ALT TRK PREVIOUSLY ASSIGNED

Cause: The HA and R0 areas designated are not defective. The alternate track was previously assigned; therefore, data will be transferred to a new alternate track.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8224I HA AND R0 OF ALT TRK IS DEFECTIVE

Cause: The HA and R0 area of the previously assigned alternate track is defective. The data portion of R0 will not be transferred, but other records may be recovered.

System Action: Processing continues.

Programmer Action: Recreate the file on another disk pack or data cell and recreate the file.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.

Operator Action: Not applicable.

8225I DATA CHECK IN COUNT FIELD

Cause: A data check has occurred in the count field. The record is not transferred to the alternate track.

System Action: Processing ccntinues.

Programmer Action: If this error affects the use of this file, recreate the file on another disk pack or data cell, or initialize this disk pack or data cell and recreate the file.

Operator Action: If the problem recurs, issue the ROD command, execute EREP, and have the output available to complete your problem determination action.

8226I NO ADDRESS MARKER

Cause: An address marker is missing. The record is not transferred to the alternate track. The pack was not properly intialized before the file was built.

System Action: Processing ccntinues.

Programmer Action: Recreate the file on another disk pack or data cell, or initialize this disk pack or data cell and recreate the file.

Operator Action: If the problem recurs, issue the ROD command, execute EREP, and have the output available to complete your problem determination action.

8227I KEY AND DATA ERROR RECOVERED

Cause: The key and data portion of this record was recovered, but is possibly in error. The record is formatted as read.

System Action: Processing continues.

Programmer Action: Check the key and data portion of the records transferred. If the problem recurs, the file should be recreated on another disk pack or data cell, or this disk pack or data cell should be initialized and the file recreated.

Note: The value in Register 1, plus 9 and 10, is the failing cylinder and head.

Operator Action: Rerun the job.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.
3. Use the address supplied by your installation programmer in the disk or data cell to printer utility to display the failing track and have the output available.

8228I KEY AND DATA ERROR

Cause: The key and data portion of this record cannot be recovered. The record is formatted with EBCDIC [A] fill characters.

System Action: Processing continues.

Programmer Action: Create the file on another disk pack or data cell, or

Initialize this disk pack or data cell and recreate the file.

If the problem recurs have the log sheet and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8229I KEY MAY BE IN ERROR

Cause: There is a possible error in the recovered key. The data field was not recovered. The record is formatted as read with the data field filled with EBCDIC [A] characters.

System Action: Processing continues.

Programmer Action: Check the key fcr validity. Create the file on another disk pack or data cell, or

Initialize this disk pack or data cell and recreate the file.

If the problem recurs, do the fcllowing to complete your problem determination action:

1. Have the log sheet and printer output available.
2. Execute a stand-alone dump at the time of the failure and have the output available.
3. Use the disk or data cell to printer utility to display the failing track.

Note: Use the dump to find the track in error. The value in register 1, plus 9 and 10, is the failing cylinder and head.

Operator Action: Not applicable.

8230I UNRECOVERABLE ERROR

Cause: An unrecoverable error has occurred, other than missing address marker, data check, or record overflow.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Rerun the job and assign another pack.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.

2. Issue the ROD command, execute EREP, and save the output.

8231I CYLxx, TRKxx, IS DEFECTIVE, AN ALTERNATIE IS ASSIGNED

Cause: The track is permanently defective and an alternate is assigned.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8232I CYLxx, TRKxx, IS NOT DEFECTIVE

Cause: The track is acceptable.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8233I CYLxx, TRKxx, HA AND R0 ARE DEFECTIVE, NO ALTERNATE ASSIGNED

Cause: The HA and R0 portion of the track is defective. An alternate track cannot be assigned.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: If the prcblem persists, the file should be recreated on another disk pack or data cell, or this disk pack or data cell should be initialized and the file recreated.

Operator Action: Rerun the job.

If the problem recurs, do the following to complete your prcblem determination action:

1. Have the job stream, log sheet, and printer output available.

2. Issue the ROD command, execute EREP, and save the output.

3. Execute a disk or data cell to printer utility for the record in question and save the output.

8234I UNRECOVERABLE DISK ERROR

Cause: An unrecoverable disk error occurred while performing surface analysis.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: Recreate the file on another disk pack or data cell, or initialize this disk pack or data cell and recreate the file.

Operator Action: Rerun the job.

If the problem recurs, do the fcllowing to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.

2. Issue the ROD command, execute EREP, and save the output.

8235I DATA TRANSFERRED TO ORGINAL DEFECTIVE TRACK

Cause: The track was found acceptable, and the data was returned to the original track.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8236I DATA TRANSFERRED TO ORIGINAL ALTERNATE TRACK

Cause: The alternate track is acceptable, and data was transferred.

System Action: Processing ccntinues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8240I END OF ALT. TRK. ASSGN

Cause: Normal end-of-job indication.

System Action: Normal system processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8250I END OF ALT. TRK. AND UPDATE

Cause: Normal end of job.

System Action: Normal system processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8251I TRACK PARAMETER MISSING

Cause: UY was specified on the utility modifier card, but TRACK= was not in the first six columns of the track statement.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Insert the TRACK statement or specify UN. Resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8252I INVALID TRACK STATEMENT

Cause: The characters in the TRACK statement are not valid hexadecimal characters, or the TRACK statement and data are missing entirely (with UY specified), or the quantity of data supplied is less than the amount specified on the TRACK statement.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the invalid TRACK statement and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8253I UPDATE RECORD CAUSES TRACK OVERFLOW

Cause: Track capacity was exceeded when attempting to write update record. The last record on the track may now be invalid.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Check the validity of the record that caused the overflow. If subsequent use of the file will be affected, the invalid record will have to be removed or the file recreated.

If the problem recurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute the file to printer utility for the record in question and have the output available.

Operator Action: Not applicable.

8256I NOT AN EMULATOR PACK

Cause: An UPSI 00000001 card was present, and the pack to be cleared was not initialized for emulator use.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Remove or correct the UPSI card if the pack is not to be used for an

emulator,or

Initialize the pack correctly for emulator use.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Check that the correct disk pack is mounted. If correction is necessary, rerun the job.

8301I SYS000 NOT ASSIGNED TO A 2321

Cause: A data cell was not assigned to SYS000.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Correct the job stream so that SYS000 is assigned to a data cell and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Assign SYS000 to a data cell and rerun the job.

8302I UNRECOVERABLE DATA CELL ERROR

Cause: An unrecoverable data cell error occurred while perfcrming a surface analysis.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Mount another data cell and rerun the job.

If the problem recurs, do the following to complete your prcblem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.

8303I CELLxx, SUBCELLxx, STRIPxx, CYLxx, TRKxx IS A DEFECTIVE ALTERNATE TRACK

Cause: The alternate track is defective and will not be assigned.

System Action: Processing ccntinues.

Programmer Action: not applicable.

Operator Action: Not applicable.

8304I CELLxx, SUBCELLxx, STRIPxx, CYLxx, TRKxx IS DEFECTIVE AND AN ALTERNATE IS ASSIGNED

Cause: The main area of the track is defective and an alternate is assigned.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8305I CELLxx, SUBCELLxx, STRIPxx, CYLxx, TRKxx IS DEFECTIVE AND NO ALTERNATE IS AVAILABLE

Cause: The track is defective and no more alternate tracks are available.

This is possibly a hardware error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Mount another data cell and rerun the job.

If the problem persists, do the fcllowing to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Issue the ROD command, execute EREP, and save the output.

8306I CELLxx, SUBCELLxx, STRIPxx, CYLxx, TRKxx HA OR R0 IS IN ERROR

Cause: The HA or R0 portion of the track is defective.

This is possibly a hardware error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Mount another cell and rerun the job.

If the problem persists, do the following to complete you problem determination action:

1. Have the job stream, log sheet, printer output available.
2. Issue the ROD command, execute EREP, and save the output.

8307I END OF INIT DATA CELL

Cause: Normal end of initialize data cell program.

System Action: Normal system processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8308I SYS000 NOT A VALID DATA CELL

Cause: The device assigned to SYS000 is not a data cell.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Execute LISTIO to get the current assignments, assign SYS000 to a data cell, and rerun the job.

If the problem recurs, have the LISTIO output available tc complete your problem determination action.

8309I CELLxx, SUBCELLxx, STRIPxx, CYLxx, TRKxx REC0 IS IN ERROR

Cause: The portion of the track where record 0 data field is written is defective.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: If the problem persists, the file should be recreated on another disk pack or data cell, or this disk pack or data cell should be initialized and the file recreated.

Operator Action: Rerun the job.

If the problem persists, do the following to complete your problem determination action:

1. Issue the ROD command, execute EREP, and have the output available.
2. Have the log sheet and printer output available.

8310I CELLxx, SUBCELLxx, STRIPxx, CYLxx, TRKxx, IS DEFECTIVE, ALTERNATE IS ASSIGNED

Cause: The track is permanently defective and an alternate is assigned.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8311I CELLxx, SUBCELLxx, STRIPxx, CYLxx, TRKxx IS NOT DEFECTIVE

Cause: The track is acceptable.

System Action: Processing continues.

Programmer Action: Not applicable.

Operator Action: Not applicable.

8312I CELLxx, SUBCELLxx, STRIPxx, CYLxx, TRKxx HA AND R0 ARE DEFECTIVE, NO ALT. ASSIGNED

Cause: The HA and R0 portion of the track is defective and an alternate track cannot be

assigned.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: If the problem recurs, the file should be recreated on another disk pack or data cell, or this disk pack or data cell should be initialized and the file recreated.

Operator Action: Rerun the job.

If the problem recurs, do the following to complete your problem determination action:

1. Issue the ROD command, execute EREP, and have the output available.
2. Have the log sheet and printer output available.

## Initialize Tape Message

8601A MORE PASSES NEEDED -- INITP

Cause: If the CARD and REWIND parameters are omitted, the program waits for a reply to initialize another tape.

System Action: The system waits for an operator response.

Programmer Action: Not applicable.

Operator Action: Reply YES if you want to initialize another tape. Reply NO or (B) to terminate the job.

Note: See Appendix D for more initialize tape messages.

## VTOC Display Messages

8V91I NO FORMAT 4 LABEL FOUND. JOB CANCELLED

Cause: The VOL1 label (cylinder 0, head 0, record 3) contains the address (cylinder, track, and record) of the Format 4 label. A record has been found at this address, but is not a Format 4 label.

This is probably a hardware error.

System Action: The job is cancelled.

Programmer Action: Assign another disk pack or data cell, or initialize this disk pack or data cell and resubmit the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

Operator Action: Not applicable.

8V92I NO VOLUME 1 LABEL FOUND. JOB CANCELLED

Cause: The VOL1 label is cylinder 0, head 0, record 3, for a disk, or subcell 0, strip 0, cylinder 0, track 0, record 3 for a data cell, when the device has been initialized. A record has been found at this address, but it is not the VOL1 label.

This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Check the device assignment and that the correct disk pack or data cell has been mounted. If correction is necessary, rerun the job. If the assignment and device are correct, initialize the disk pack or data cell and rerun the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

8V93I INVALID VTOC ADDR FOUND. JOB CANCELLED

Cause: The VTOC address in the VOL1 label is less than cylinder 0, head 0, record 4, or is equal to or greater than cylinder 200, head 0, record 0. This is probably a user error.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Assign another disk pack or initialize this disk pack and rerun the job.

If the problem recurs, have the job stream, log sheet, and printer output available to complete your problem determination action.

8V94I NO DISK RECORD FOUND. JOB CANCELLED

Cause: A disk pack has been mounted that was not initialized or a hardware error has occurred.

System Action: The job is cancelled.

Programmer Action: Not applicable.

Operator Action: Check that the pack has been initialized. If not, initialize it and rerun the job. If the pack has been initialized, move it to another drive and rerun the job. If the job executes normally, the original drive is malfunctioning.

For a disk drive malfunction or persistant errors, issue the ROD command, execute EREP, and have the output available to complete your problem determination action.

8V95I NOT A VALID LABEL FORMAT

Cause: A label other than type 1, 2, 3, 5, or X'00' was encountered after the Format 4 label was processed.

System Action: Processing continues.

Programmer Action: Check byte 45 of all labels for one of the valid format identifiers.

Operator Action: Execute the LISTVTOC program and give the output to the programmer.

8V96D FORMAT 1 LABEL OF DATA SECURED FILE

Cause: A Format 1 label describing a data secured file has been read by the VTOC Display utility program.

System Action: The system waits for an operator response.

Programmer Action: Instruct the operator what to reply when this message is issued.

If a failure occurs, do the following to complete your problem determination action:

1. Have the job stream, log sheet, and printer output available.
2. Execute a stand-alone dump at the time of the failure.

Operator Action: Reply YES to allow the label information of all data secured files to be listed along with the rest of the label information in the VTOC.

Reply NO or Ⓑ to allow only the label information pertaining to unsecured files to be listed.

Any other response results in an invalid response message.