

FATS®  
FATS  
FATS  
FATS  
FATS  
FATS  
FATS

# V4.3

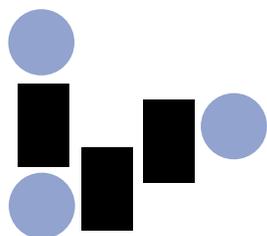
## FATS/FATAR USER DOCUMENTATION

### FATS®

Fast Analysis of Tape Surfaces

### FATAR™

Fast Analysis of Tape And Recovery



**INNOVATION®**  
**DATA PROCESSING**

**CORPORATE HEADQUARTERS:** 275 Paterson Ave, Little Falls, NJ 07424 • (973) 890-7300 • Fax: (973) 890-7147  
E-mail: [support@fdrinnovation.com](mailto:support@fdrinnovation.com) • [sales@fdrinnovation.com](mailto:sales@fdrinnovation.com) • <http://www.innovationdp.fdr.com>

|                          |                          |                         |                             |                                 |                                     |
|--------------------------|--------------------------|-------------------------|-----------------------------|---------------------------------|-------------------------------------|
| <b>EUROPEAN OFFICES:</b> | FRANCE<br>01-49-69-94-02 | GERMANY<br>089-489-0210 | NETHERLANDS<br>036-534-1660 | UNITED KINGDOM<br>0208-905-1266 | NORDIC COUNTRIES<br>+31-36-534-1660 |
|--------------------------|--------------------------|-------------------------|-----------------------------|---------------------------------|-------------------------------------|



# FATS<sup>®</sup> & FATAR<sup>™</sup>

USER DOCUMENTATION



**1.0 INTRODUCTION**

FATS (Fast Analysis of Tape Surfaces) and FATAR (Fast Analysis of Tape and Recovery) are magnetic tape utility programs which provide a number of functions, including certification and verification of tape volumes, erasing tapes, labeling tapes, analyzing tapes and copying tapes.

FATS and FATAR are proprietary products of:

INNOVATION DATA PROCESSING, INC.

Innovation Plaza  
275 Paterson Avenue  
Little Falls, NJ 07424  
Tel: (201) 890-7300  
Fax: (201) 890-7147

FATS and FATAR are trademarks of INNOVATION DATA PROCESSING and are registered with the US Patent Office.

FATS and FATAR are available for license exclusively from INNOVATION DATA PROCESSING, INC.

**EUROPEAN OFFICES:****France**

Innovation Data Processing S.A.R.L.  
66 Avenue François Arago  
92000 Nanterre  
Tel: 01-47-69-15-15 Fax: 01-47-69-15-16  
E-mail: 76543.1377@compuserve.com

**Netherlands**

Innovation Data Processing  
Brouwerstraat 8  
1315 BP Almere  
Tel: 036-534 1660 Fax: 036-533 7308  
E-mail: 76543.1376@compuserve.com

**Germany**

Innovation Data Processing  
International Ltd.  
Orleansstraße 4a  
D-81669 München  
Tel: 089-489 0210 Fax: 089-489 1355  
E-mail: 76543.1401@compuserve.com

**United Kingdom**

Innovation Data Processing Ltd.  
Clarendon House  
125 Shenley Road  
Borehamwood, Herts, WD6 1AG  
Tel: 0208-905 1266 Fax: 0208-905 1428  
E-mail: 76543.1374@compuserve.com

**Foreign Representatives:**

Computer Supervisory  
Services  
4 Brandon Street  
Clovelly NSW  
Australia 2031  
Tel: 61-2-665-1104  
Fax: 61-2-665-7382

Eckert Research International  
Corporation  
2-7-10 Nihonbashi Kayabacho  
Chuo-ku, Tokyo 103  
Japan  
Tel: 03-668-8866  
Fax: 03-668-8874

Stratos--Strategic Tools &  
Services  
Via Assarotti, 9  
10122 Torino, Italy  
Tel: 11-5620 354  
Fax: 11-5621 291

© Copyright 1995, 1996 INNOVATION DATA PROCESSING, INC.

**Reproduction of this User Manual is prohibited except for licensed FATS and FATAR customers.**

Main Phone Number (US)  
201-890-7300

Fax Number (US)  
201-890-7147

E-mail (US): 76322.2076@compuserve.com  
<http://www.innovationdp.fdr.com>

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

# TABLE OF CONTENTS

PAGE NO.

|           |  |           |
|-----------|--|-----------|
| <b>01</b> | <b>INTRODUCTION</b>                          |           |
| 01.1      | SUMMARY OF NEW FATS AND FATAR FEATURES ..... | v         |
| <b>10</b> | <b>FATS FUNCTIONAL DESCRIPTION</b> .....     | <b>1</b>  |
| <b>11</b> | <b>FATS TECHNICAL SUMMARY</b> .....          | <b>3</b>  |
| 11.1      | GENERAL .....                                | 3         |
| 11.2      | TAPE CERTIFICATION .....                     | 4         |
| 11.3      | TAPE VERIFICATION .....                      | 7         |
| 11.4      | TAPE LABELING .....                          | 8         |
| 11.5      | ERASING TAPES .....                          | 9         |
| 11.7      | TAPE MANAGEMENT SYSTEMS .....                | 10        |
| 11.8      | REPORT FORMATS .....                         | 12        |
| 11.10     | IDRC OVERVIEW .....                          | 14        |
| 11.11     | 3490E OVERVIEW .....                         | 16        |
| <b>12</b> | <b>FATS GUIDELINES</b> .....                 | <b>19</b> |
| 12.1      | CAUSES OF TAPE ERRORS .....                  | 19        |
| 12.2      | OPERATING SYSTEM VS. FATS ERRORS .....       | 19        |
| 12.3      | VARIATIONS IN FATS RESULTS .....             | 20        |
| 12.4      | ANALYSIS OF CERTIFICATION RESULTS .....      | 21        |
| 12.5      | ANALYSIS OF VERIFICATION RESULTS .....       | 22        |
| 12.6      | RECOMMENDATIONS FOR FATS USAGE .....         | 22        |
| 12.7      | QUESTIONS AND ANSWERS .....                  | 23        |
| 12.8      | CERTIFICATION OF TAPE CARTRIDGES .....       | 24        |
| <b>13</b> | <b>FATS EXECUTION JCL</b> .....              | <b>25</b> |
| 13.1      | OPERATOR CONTROL .....                       | 27        |
| <b>14</b> | <b>FATS CONTROL STATEMENTS</b> .....         | <b>29</b> |
| 14.1      | GENERAL .....                                | 29        |
| 14.2      | OPERATION KEYWORDS .....                     | 30        |
| 14.3      | OPTION PARAMETERS .....                      | 31        |
| 14.3.1    | RETRY/ERROR PARAMETERS .....                 | 31        |
| 14.3.2    | VOLUME MOUNTING PARAMETERS .....             | 32        |
| 14.3.3    | LABELING PARAMETERS .....                    | 34        |
| 14.3.4    | PRINT PARAMETERS .....                       | 35        |
| 14.3.5    | CONTROL PARAMETERS .....                     | 36        |
| 14.3.6    | MISCELLANEOUS PARAMETERS .....               | 37        |
| 14.3.7    | ANALYZE PARAMETERS .....                     | 38        |
| <b>16</b> | <b>FATS EXAMPLES OF USAGE</b> .....          | <b>39</b> |
| 16.1      | INDEX TO EXAMPLES .....                      | 39        |
| 16.2      | EXAMPLES .....                               | 40        |

## TABLE OF CONTENTS

|   | PAGE NO.  |
|---|-----------|
| <b>20 FATAR FUNCTIONAL DESCRIPTION</b> .....      | <b>45</b> |
| <b>21 FATAR TECHNICAL SUMMARY</b> .....           | <b>48</b> |
| 21.1 GENERAL .....                                | 48        |
| 21.2 INPUT TAPE PROCESSING .....                  | 48        |
| 21.3 MODIFICATION AND SCANNING .....              | 51        |
| 21.4 OUTPUT TAPE PROCESSING .....                 | 52        |
| <b>22 FATAR OPERATIONS</b> .....                  | <b>53</b> |
| 22.1 INPUT TAPES AND LABELS .....                 | 53        |
| 22.2 MODIFICATION AND SCANNING .....              | 55        |
| 22.3 OUTPUT TAPES .....                           | 55        |
| <b>23 FATAR EXECUTION JCL</b> .....               | <b>56</b> |
| <b>24 FATAR CONTROL STATEMENTS</b> .....          | <b>59</b> |
| 24.1 GENERAL .....                                | 59        |
| 24.2 ANALYZE CONTROL STATEMENT .....              | 60        |
| 24.2.1 INPUT TAPE CONTROL KEYWORDS .....          | 62        |
| 24.2.2 ERROR RECOVERY CONTROL KEYWORDS .....      | 63        |
| 24.2.3 OUTPUT TAPE CONTROL KEYWORDS .....         | 64        |
| 24.2.4 PRINT CONTROL KEYWORDS .....               | 65        |
| 24.2.5 MISCELLANEOUS KEYWORDS .....               | 66        |
| 24.3 MODIFICATION/SCAN CONTROL STATEMENTS .....   | 67        |
| 24.3.1 PRINT STATEMENT .....                      | 67        |
| 24.3.2 DROP STATEMENT .....                       | 67        |
| 24.3.3 KEEP STATEMENT .....                       | 68        |
| 24.3.4 SCAN STATEMENT .....                       | 68        |
| 24.3.5 REPLACE STATEMENT .....                    | 69        |
| 24.3.6 TABLE STATEMENT .....                      | 69        |
| 24.4 MODIFICATION/SCAN PARAMETERS .....           | 70        |
| <b>25 FATAR REPORT EXAMPLES</b> .....             | <b>72</b> |
| <b>26 FATAR EXAMPLES OF USAGE</b> .....           | <b>75</b> |
| 26.1 INDEX TO EXAMPLES .....                      | 76        |
| 26.2 EXAMPLES .....                               | 77        |
| <b>30 FATS AND FATAR MESSAGES AND CODES</b> ..... | <b>93</b> |
| 30.1 RETURN CODES .....                           | 93        |
| 30.2 ABEND CODES .....                            | 93        |
| 30.3 CONSOLE MESSAGES .....                       | 94        |
| 30.4 PRINT MESSAGES .....                         | 97        |
| 30.5 FATS ACTION MESSAGES .....                   | 110       |

## TABLE OF CONTENTS

|   | PAGE NO.   |
|---|------------|
| <b>40 SPECIAL CONSIDERATIONS</b> .....      | <b>111</b> |
| 40.1 VARIABLE SPANNED RECORD ABENDS .....   | 111        |
| 40.1.1 CORRECTING S002 ABENDS .....         | 112        |
| <b>90 FATS AND FATAR INSTALLATION</b> ..... | <b>113</b> |
| 90.0 FATS AND FATAR INSTALLATION .....      | 113        |
| 90.02 PRINTING THE USER MANUAL .....        | 119        |
| <b>93 FATS AND FATAR ISPF PANELS</b> .....  | <b>121</b> |

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## 1.1 SUMMARY OF NEW FEATURES

The following are new features of FATS V4.3.

ISPF panels are available to perform many of the FATS/FATAR functions like labeling, certifying and copying tapes.

The panels simplify the use of FATS/FATAR for common tasks like certifying or labeling tapes. Extensive editing of the input is done to eliminate control card errors at execution time. More complicated tasks such as dropping blocks or replacing data on a tape will be easier to perform.

[See Section 93 for documentation.](#)

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

# FATS<sup>®</sup> & FATAR<sup>™</sup>

USER DOCUMENTATION

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## 10.0 FATS FUNCTIONAL DESCRIPTION

**WHAT IS FATS?** FATS (Fast Analysis of Tape Surfaces) is a special-purpose magnetic tape utility, which can be executed under MVS, MVS/XA, and MVS/ESA. FATS is used to perform many of the functions for which a tape librarian is responsible in most shops, such as certifying usability of new or scratch tapes, verifying readability of tapes, erasing data from tape, and labeling tapes. However, these operations are done on the actual tape drives where the tapes will be used rather than on expensive Stand-Alone hardware devices. This results in a savings of money and of time, since FATS can perform as many concurrent operations as there are tape drives available.

**WHAT IS FATS USED FOR?** FATS has several specific applications, including:

- Certifying the usability of tapes, both new and old, by writing a pattern on the full length of the tape (this is equivalent to the function of hardware tape certifiers). This helps to avoid write data errors at application run time by identifying potentially defective tapes.
- Verifying the readability of critical or archival data tapes, to avoid read data errors at application run time. Tapes with read errors may be recovered using FATS' companion product FATAR.
- Labeling tapes with specified volume serials. This can be done by itself, or in combination with a certification process.
- Erasing data from a tape, to insure that confidential information is not left on a tape (as when a tape volume is being sold or shipped to another location). This is similar to stand-alone tape degaussers, but is superior since tape serials can be verified and preserved.
- Interfacing with its companion product FATAR to perform detailed data analysis and copying of tapes.

Regular use of the certification and verification functions of FATS can significantly reduce the number of application abends due to data checks. In addition, many tape data checks are recoverable by operating system error recovery, but cause significant run time degradation while this recovery is taking place; FATS can reduce this degradation.

**WHO USES FATS?** FATS is normally used by tape librarians and operations personnel but may be used by systems and applications programmers.

**TAPE OPERATIONS** Depending on the command statements provided, FATS will perform any of its supported functions on from 1 to 9 tape drives simultaneously. The same operation could be performed on all of the allocated tape drives, or, if desired, different functions could be done on different drives. FATS uses an internal task control so that all of these operations can proceed independently of one another. by the use of FATS parameters, FATS can be instructed to request multiple tapes on a drive, calling either for scratch tapes or specific volume serials, so that the same operation can be performed on many tapes using one tape drive. FATS will terminate when instructed by the operator or when all requested functions are completed.

CONTINUED . . .

**10.0 CONTINUED . . .**

**SUPPORTED DEVICES** FATS will function on any density drive up to and including 6250 BPI, and also on the IBM 3480, 3490 and 3490E tape cartridge systems. By default, 9-track and 3480/3490 tapes will be certified at the highest density possible on the tape drive, although, for drives capable of multiple densities, this can be overridden. Certification or verification of 7-track tapes requires specification of a "MODE=" parameter giving density and other options. All IBM and IBM-compatible tape drives are supported by FATS; non-supported tape drives can also be handled by FATS by specification of their density (BPI) and inter-record gap size.

**PROCESSING SPEED** The speed at which FATS can process a tape is dependent on three variables: the speed of the tape drive, the contention for the tape channel and control unit, and the quality of the tapes being tested. When testing tapes suspected of being exceptionally poor, a low permanent error retry level can be specified to accelerate processing. FATS may also contend with itself if tapes being tested are on the same tape control unit or channel; when processing more than three tapes in one FATS job (except for "ERASE" functions), it is wise to allocate more than one channel and control unit, if available.

## 11.0 FATS TECHNICAL SUMMARY

### 11.1 GENERAL

**THE FATS PROGRAM** FATS (Fast Analysis of Tape Surfaces) is a utility program for certifying, verifying, labeling, and erasing magnetic tapes.

FATS requires 256K of memory. However, if the default tape certification blocksize is overridden to a value greater than 32760, then that blocksize will be acquired via a GETMAIN and must be added to the memory requirement. If FATS' companion product FATAR is invoked as a sub-task (via the "ANALYZE(n)" operation), then the memory requirements of FATAR (detailed elsewhere in this manual) must be added to that of FATS for each FATAR sub-task.

Although FATS is normally linkedited as an authorized program it requires authorization only if the "BLP" control statement keyword is used, if 3480 tape cartridge drives are to be used, or if it is desired to take advantage of FATS/FATAR's ability to automatically reply to CA-1 (UCC1, TMS) messages.

**CONTROL AND REPORT I/O** FATS accepts its control statement input from DD statement "SYSIN". The control statements are 80-byte records, of which only columns 1 to 71 may contain information.

Report output is directed to several print datasets. DD statement "SYSPRINT" is required, and will contain general messages and control statement listings. Up to 9 optional "SYSPRINx" DD statements may be included (where "x" is a digit from 1 to 9 and corresponds to the 9 tape drives used by FATS); if present, detail reports on the results from the associated tape drive will be printed on "SYSPRINx", but if omitted, the detail reports will go to "SYSPRINT". DD statement "SUMMPRT" is also optional; if present, it will receive a summary report on each tape volume processed by FATS; if absent, the summary report will go to the associated "SYSPRINx" or to "SYSPRINT".

**TAPE OPEN PROCESSING** FATS always opens every tape with a standard data management OPEN, so that all label processing (if not bypassed in the JCL) will be performed, and tape management and security systems, if present, will be invoked. So, for all FATS operations, you have assurance that the tapes mounted are the proper tapes and that they may be written on or read from as long as you do not bypass these checks yourself.

If label processing is bypassed ("LABEL=(,BLP)" in JCL), the FATS option "VALIDATE=" may be used to check the volume serial and expiration date on tapes used by FATS.

**FATS CONTROLS** The system operator or tape librarian can exercise control over FATS by optionally replying to an outstanding WTOR (write-to-operator-with-reply) or by issuing an OS STOP or MODIFY command. Use of WTOR=YES operand (default) will cause a WTOR to be issued to the tape pool console (route code 3) when FATS is executing. The operator or librarian can terminate FATS at any time by replying to this message; FATS can also be instructed to abandon processing of a particular tape (if, for example, the tape is too damaged to mount). Alternatively, by using the MODIFY=YES parameter (default is MODIFY=NO), the operator or librarian may use OS STOP and MODIFY commands to gain the same control over FATS.

**ISPF PANEL SUPPORT** Extensive ISPF Panels are available to perform many of the FATS/FATAR functions like labeling, certifying and copying tapes. [See Section 93](#) on "How to Use the ISPF Panels."

## 11.2 TAPE CERTIFICATION

### **OPERATING SYSTEM RECOVERY**

When the operating system issues a write command to a tape drive, the tape hardware reads back the data written and indicates to the operating system the success or failure of the WRITE. If the WRITE failed, and the sense data (Hardware Error flags) indicate that a data check was the cause, the operating system will enter a RETRY loop where it will:

- Backspace over the bad record.
- Erase a section of tape (3.5 inches on IBM 3420 tape drives, .31 inches on IBM 3480 tape cartridge systems).
- Retry the failing WRITE command.

This will normally be done until the write completes normally or until 15 RETRIES have been done (16 WRITE attempts in all). Note that in the IBM 3480 tape cartridge system, these RETRY operations are done by the tape control unit without assistance from the operating system, but the effect is the same.

If all RETRIES fail, the operating system will write an I/O error message on the console and report the error back to the requesting application program, which usually results in abnormal termination of that job.

Even if the recovery is eventually successful, all of the repositioning and retrying can be an expensive consumer of CPU time and can monopolize tape channels and control units. In the worst case, over 50 inches of tape may be erased to recover from a very long error; a number of these errors can significantly reduce the usable length of a tape.

Permanent (unrecoverable) and temporary (eventually successful) data check counts are accumulated by the operating system and recorded both on "SYS1.LOGREC" (EREP) and on SMF (if you are keeping SMF type 21 records which are "ESV" – Error Statistics by Volume). However, all of the above RETRY operations are recorded as only one permanent or temporary error even though the WRITE may have failed at up to 16 different locations on the tape. No indication is given of the length of an error.

The purpose of operating system error recovery is to attempt to let an application job complete successfully despite WRITE errors on output tapes, regardless of the cost. This is an excellent goal, but greater reliability and performance could be achieved if the errors could be avoided in the first place.

CONTINUED . . .

## 11.2 CONTINUED . . .

**TAPE  
CERTIFI-  
CATION**

The FATS certification function identifies error locations on a tape before it is used by an application program.

The technique used is simple. FATS writes large blocks (the size varies by tape density) using a pattern which attempts to defeat the automatic 1- and 2-BIT error correction performed by many tape drives. The hardware readback function of the tape drive is used to detect blocks which were not written successfully (data checks).

When data checks occur, FATS will backspace over the bad block and RETRY the WRITE in the same location. If a user-specified RETRY limit is exceeded for errors at one location (default is 10), the error is considered permanent. If the WRITE is completed successfully before the RETRY limit is reached, it is considered a temporary data check (this may be due to dirt on the tape which was dislodged during the RETRY process).

FATS will report, in the associated detail report for the tape, the location in feet of any temporary or permanent error. The length of permanent errors in inches will also be reported. If permanent errors occur at consecutive locations, FATS will report the cumulative length of the error. The summary report for the tape will report the total number of permanent errors, and temporary error totals by number of RETRIES.

By reporting the number, length, and location of all errors, FATS provides sufficient information to make an informed decision on whether a given tape should be put into use, discarded, or repaired ([more detail on this is given in the section "FATS GUIDELINES"](#)).

The location of errors is calculated by FATS in feet since the beginning of the tape as a function of the number and length of FATS records written, and the number of inter-record gaps. FATS assumes the nominal inter-record gap size for each tape drive and density as documented in IBM tape drive hardware manuals. However, the actual length of this gap can vary from drive to drive, so a given tape certified on two different drives may report errors at slightly different locations. FATS will also report the total length of the tape, but this may also vary due to gap variations.

Tape data checks may be caused by tape drive equipment checks, by tape surface damage (or improper manufacture), or by dirt or other contaminants on the surface. Equipment failure is the least likely and will usually show up as errors across the entire length of a number of tapes. errors caused by dirt are often transitory; they may move or disappear if the tape drive pick up the dirt and re-deposit it in a different location. A thorough cleaning of a tape with permanent errors (using an inexpensive stand-alone tape cleaner) will often correct many of the errors and reclaim a tape about to be discarded.

**CERTIFI-  
CATION OF  
TAPE  
CARTRIDGES**

Tape cartridges were first introduced in 1985 and resulted in a significant reduction in the number of data checks compared to 6250 BPI tape. FATS is an excellent tool to certify cartridges and insure that the cartridges that have been in the library for a number of years and have potential for errors be **identified** and **eliminated** from the tape library.

**Significant cost and time is required to identify the bad batch of cartridges and to copy the data to a new set of tapes.**

CONTINUED . . .

## 11.2 CONTINUED . . .

FATS Certify – This is Option A from the FATS/FATAR Main Menu. [See Section 93](#) for further information.

**FATS CERTIFY  
NEW TAPES**

```

----- FATS CERTIFY NEW TAPES -----
COMMAND ===> _

TAPE1 DD Dsname=  FATS                               <
          Unit  =(  TAPE                               )
          Label =(  ,BLP,EXPDT=98000                    )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                          5: _____  6: _____  7: _____  8: _____
                                          9: _____ 10: _____ 11: _____ 12: _____
                                          13: _____ 14: _____ 15: _____ 16: _____
                                          17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
NOLABEL -- CREATE no label (NL) tapes: NO (yes/no)
OWNERID  -- Contents of owner ID field in output volume label: _____

```

FATS certification of existing tapes whose data sets have expired. This is Option B from the FATS/FATAR Main Menu. [See Section 93](#) for further information.

**FATS CERTIFY  
EXISTING  
TAPES**

```

----- FATS CERTIFY EXISTING TAPES -----
COMMAND ===> _

TAPE1 DD Dsname=  FATS                               <
          Unit  =(  TAPE                               )
          Label =(  ,SL                                )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                          5: _____  6: _____  7: _____  8: _____
                                          9: _____ 10: _____ 11: _____ 12: _____
                                          13: _____ 14: _____ 15: _____ 16: _____
                                          17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
NOLABEL -- CREATE no label (NL) tapes: NO (yes/no)
OWNERID  -- Contents of owner ID field in output volume label: _____

```

### 11.3 TAPE VERIFICATION

#### TAPE VERIFICATION

When reading tape, the operating system will retry any read data check up to 40 times by backspacing and re-reading the block. Cleaning actions (attempts to dislodge any dirt by moving the tape rapidly back and forth) are done periodically. In the end, if the tape drive is unable to read the block successfully, the error is reported to the application program which usually abnormally terminates.

FATS read verification may be used on critical or archival tapes before they are required by an application job to verify their readability. FATS will read each block on the input file, and will attempt to re-read each data check block a user-specified number of times (10 retries by default).

FATS will report in its detail report the exact location (in feet), file number, and record number of any temporary or permanent read data check encountered. Much like the certification function, FATS will report the length of any permanent error, and the total length of any consecutive permanent errors. The summary report for the verification will show the number of permanent errors, and the number of temporary errors by the number of retries.

FATS performs its verification by reading without actually transferring the data read to the CPU; this is sufficient to identify data checks. For a more detailed look at the data causing the errors, or to attempt to correct or bypass the errors, FATS' companion product FATAR may be used. FATAR can be executed directly, or as a sub-task of FATS. FATAR is more appropriate for verifying tapes where the number of data files on the tape is unknown.

FATS verify (read) of existing data on a tape. This is Option F of the FATS/FATAR Main Menu. [See Section 93](#) for further information.

#### FATS VERIFY (READ) OF EXISTING DATA ON TAPE

```

----- FATS VERIFY (READ) TAPES -----
COMMAND ==> _

TAPE1 DD Dsname= FATS
Unit =( TAPE )
Label =( ,BLP,EXPDT=98000 )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                           5: _____  6: _____  7: _____  8: _____
                                           9: _____ 10: _____ 11: _____ 12: _____
                                           13: _____ 14: _____ 15: _____ 16: _____
                                           17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
FILES   -- Number of Physical files to be read: 3

```

## 11.4 TAPE LABELING

FATS labels a tape by writing an IBM standard VOL1 label LABELING with a specified volume serial and (optionally) an owner name. It also writes header and trailer labels and tape marks so that the tape is a valid standard labeled tape.

The HDR1/EOF1 labels will contain whatever data set name appeared on the TAPEX DD statement except that if "LABEL=EXPDT=98000" appears the data set name will consist of 17 "0" so that the tape will be an acceptable scratch tape for tape management systems. The data set creation and expiration dates will be set to the current date. The HDR2/EOF2 labels will show DCB parameters of RECFM=U,BLKSIZE=32760.

FATS can write labels during certification of a tape, and can also label tapes as a separate function (at considerable savings in elapsed time). Because of FATS abilities to request mounting of a large number of volumes, it is ideal for initializing a sequence of new tapes, or re-labeling old tapes with new serials.

The serial number can be specified in FATS control statements, in JCL or via the operator's console. However, for ease in certifying tapes which are already labeled, the FATS default option "SAVLAB" will read the existing volume serial from the tape being certified and re-label the tape with it, thus "saving" the volume serial.

FATS labeling of new tapes. This is Option C of the FATS/FATAR Main Menu. [See Section 93](#) for further information.

### FATS LABELING OF NEW TAPES

```

----- FATS LABEL NEW TAPES -----
COMMAND ==> _

TAPE1 DD Dsname= FATS                               <
Unit  =( TAPE )
Label =( ,BLP,EXPDT=98000 )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                           5: _____  6: _____  7: _____  8: _____
                                           9: _____ 10: _____ 11: _____ 12: _____
                                           13: _____ 14: _____ 15: _____ 16: _____
                                           17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
NOLABEL -- CREATE no label (NL) tapes: NO (yes/no)
OWNERID  -- Contents of owner ID field in output volume label: _____

```

## 11.5 ERASING TAPES

### ERASING TAPES

Most IBM and IBM-compatible tape drives support a "Data Security Erase" feature, which is used to erase data on a tape at high-speed without tying up the tape control unit. In other words, this erase function can clear all data from a tape without impacting other jobs using tape drives on the same tape control unit. The erase takes place at maximum tape write speed (which varies by tape drive model).

"Data security Erase" is not supported by any standard IBM software. However, FATS fully supports the erase function. When invoked, FATS ERASE will write labels on the tape (if indicated), and then start the hardware erase function which will continue without intervention from the CPU or control unit until the end-of-tape reflector is reached. For 3480 tape cartridge systems, erase involves writing a random pattern on the tape for complete security. For other tape systems, the tape drive's erase head is used to erase the tape to "unrecorded" status.

This ERASE function may be important to installations which are upgrading to the IBM 3480 tape cartridge drives and are planning to sell many of their old 1/2 inch tapes. **ERASE may be used to insure that all company confidential data is gone from those reels before they leave the site.**

Note that the ERASE function, because it is a single very long I/O operation, may receive messages from the operating system indicating that "Device End is Missing" on the tape drive; this is normal and can be ignored. Also note that IBM 3420 tape drives (and some IBM-compatible drives) will not light the "SELECT" light while the tape is being erased; operators need to be aware that this condition may exist. IBM 3480 tape cartridge systems will display the message "ERASE" or "ERASING" while the ERASE operation is in progress.

FATS Erase Function. This is Option E from the FATS/FATAR Main Menu. [See Section 93](#) for further information.

**WARNING: This option will erase all data on a tape/cartridge. Use caution when using this option.**

### FATS ERASE FUNCTION

```

----- FATS ERASE OLD DATA -----
COMMAND ==> _

TAPE1 DD Dsname= FATS                               <
Unit  =( TAPE      )
Label =(      ,SL      )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                           5: _____  6: _____  7: _____  8: _____
                                           9: _____ 10: _____ 11: _____ 12: _____
                                           13: _____ 14: _____ 15: _____ 16: _____
                                           17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
SAVLAB  -- Preserve volume serial?   : YES (yes/no)
NOLABEL -- CREATE no label (NL) tapes: NO (yes/no)
OWNERID -- Contents of owner ID field in output volume label: _____

```

## 11.7 TAPE MANAGEMENT SYSTEMS

### **TAPE MANAGE- MENT SYSTEMS**

Although FATS has no formal interface to any tape management system, it is designed to be compatible with them. FATS takes the following actions for the benefit of tape management systems:

- Tapes will always be opened with the label type indicated in the JCL (unless the FATS "BLP" option is used). If tapes are opened as labeled, the tape management system knows the volume serial of the tape and can approve or disapprove its use. If they are opened as unlabeled, an operator response may be required to provide the volume serial for the tape management system.
- When tapes are opened for output (certification), the expiration date of the dataset will be set to the current date, so that the dataset will be immediately available as a scratch tape.
- If labels are written on the tape, the dataset name will be the name given in the JCL for the tape, unless EXPDT=98000 is given when a dataset name of 17 "0"s is used.

Most tape management systems provide the ability to bypass their operation on a particular TAPE DD Statement. This may be required for certain operations with FATS. For example, if multiple volumes are being verified (read) by FATS, and their dataset names differ, FATS will open them all with the dataset name on the TAPE DD Statement; this may fail unless the tape management system is bypassed.

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## 11.8 REPORT FORMATS

**DETAIL REPORT** The FATS detail report contains one line for each significant event which occurs on a tape being processed by FATS. Significant events include labels written or saved, permanent or temporary data checks, end-of-tape (TAPE INDICATE), tape marks read, and certain error conditions. Each detail line will include:

- The TAPE DD name (ID), e.g., TAPE1
- The tape unit address (UCB), e.g., 381
- The tape volume serial (LABEL)
- The FATS operation keyword (OPTION), e.g, WRITE
- The PASS number (for WRITE only, to differentiate the results of the two tape certifications when PASS=2)
- The file number (FILE NO) and record number (RECORDS) within the file (for READ only)
- The LOCATION, in feet from the beginning of the tape, of the event
- The LENGTH, in inches, of any permanent data check (if contiguous data checks occur, this length is cumulative)
- The number of RETRIES attempted for any data check
- An ACTION message detailing the type of event ([these messages are listed in the "Messages and Codes" section of this manual](#)).

From this detail report, it is possible to see where the errors are and how they are grouped. For a WRITE operation, the location listed for TAPE INDICATE (end-of-tape) shows the total length of the tape. For a READ operation, the line for TAPE MARK gives the number of records and length of each file (the length is cumulative, so you must subtract to get individual file lengths).

By default, FATS will print a blank line for every four detail report lines, and will print a blank line whenever a new volume is mounted on a tape unit. If the detail report is printed on a separate report file (DD name SYSPRINx rather than SYSPRINT), options are available to control the number of detail lines grouped together, and to skip to a new page for a new volume.

**SUMMARY REPORT** The FATS summary report will contain several lines for each tape volume processed by FATS. The first line will contain ID, UCB, LABEL, and option (same as in the detail report above), plus the completion code (NORMAL or ABNORM), and the total length, in feet, of the tape as processed by FATS.

The second line will appear once or twice (if two passes are done), and gives the number of data checks encountered on the tape summarized by the number of retries performed for each. Counters for retry levels 1 through 10 appear on this line; the word "\*\*PERM" appears above the column which is considered a permanent error by FATS (the value of the "RETRY=" parameter); non-zero counts for any lesser retry values indicate temporary errors. If "RETRY=" had a value greater than 10, this line will contain only the total number of temporary and permanent data checks, with appropriate headings above.

CONTINUED . . .

11.8 CONTINUED . . .

**FATS CONTROL REPORT**

FAST ANALYSIS OF TAPE SURFACES CONTROL REPORT -- FATS VER 4.0 -- INNOVATION DATA PROCESSING DATE 85.214 PAGE 0001

FATS100 FOR INSTRUCTIONS AND MESSAGES AND CODES USE PARM=I  
 FATS110 WRITE(1) OWNERID='FATS TEST',MAXERR=10,WTO, 00009407  
 FATS110 VOL=(FAT002,FAT007,000511) 00009500  
 FATS111 CHARACTERISTICS OF THE TAPES TO BE ANALYZED  
 FATS111 ID PASS OPTION LABEL FILES RETRY MODE BPI GAP REWIND THRESHOLD MAXERR BLKSIZE  
 FATS111 TAPE1 1 WRITE SAV 10 03 06250 1875 YES 01000 00010 32760  
 FATS301 END OF REPORT

**FATS DETAIL REPORT**

FAST ANALYSIS OF TAPE SURFACES DETAIL REPORT -- FATS VER 4.0 -- INNOVATION DATA PROCESSING DATE 85.214 PAGE 0001

| MESSAGE | ID            | UCB | LABEL  | OPTION | PASS | FILE NO | RECORDS | LOCATION | LENGTH | RETRIES | ACTION               |
|---------|---------------|-----|--------|--------|------|---------|---------|----------|--------|---------|----------------------|
| FATS107 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 1 FT     |        |         | LABEL SAVED          |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 5 FT     |        | 05      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 56 FT    |        | 05      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 67 FT    |        | 08      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 79 FT    |        | 01      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 104 FT   |        | 02      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 104 FT   | 5 IN   | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 105 FT   |        | 01      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 105 FT   | 5 IN   | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 107 FT   |        | 05      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 107 FT   |        | 01      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 108 FT   | 5 IN   | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 109 FT   |        | 03      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 109 FT   | 5 IN   | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 109 FT   |        | 02      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 111 FT   | 5 IN   | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 112 FT   |        | 08      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 113 FT   |        | 02      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 113 FT   | 5 IN   | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 114 FT   |        | 01      | TEMP DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 114 FT   | 5 IN   | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 115 FT   | 11 IN  | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 115 FT   | 16 IN  | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 115 FT   | 22 IN  | 10      | PERM DATA CHECK      |
| FATS207 | TAPE1         | 383 | FAT002 | WRITE  | 1    |         |         | 116 FT   | 27 IN  |         | MAX ERROR EXCEEDED * |
| FATS107 | TAPE1         | 383 | FAT007 | WRITE  | 1    |         |         | 1 FT     |        |         | LABEL SAVED          |
| FATS204 | TAPE1         | 383 | FAT007 | WRITE  | 1    |         |         | 1 FT     | 5 IN   | 10      | PERM DATA CHECK      |
| FATS204 | TAPE1         | 383 | FAT007 | WRITE  | 1    |         |         | 189 FT   | 5 IN   | 10      | PERM DATA CHECK      |
| FATS207 | TAPE1         | 383 | FAT007 | WRITE  | 1    |         |         | 198 FT   |        |         | TAPE INDICATE *****  |
| FATS107 | TAPE1         | 383 | 000511 | WRITE  | 1    |         |         | 1 FT     |        |         | LABEL SAVED          |
| FATS204 | TAPE1         | 383 | 000511 | WRITE  | 1    |         |         | 2361 FT  |        | 02      | TEMP DATA CHECK      |
| FATS207 | TAPE1         | 383 | 000511 | WRITE  | 1    |         |         | 2363 FT  |        |         | TAPE INDICATE *****  |
| FATS301 | END OF REPORT |     |        |        |      |         |         |          |        |         |                      |

**FATS SUMMARY REPORT**

FAST ANALYSIS OF TAPE SURFACES SUMMARY REPORT -- FATS VER 4.0 -- INNOVATION DATA PROCESSING DATE 85.214 PAGE 0001

| MESSAGE | ID            | UCB | LABEL  | OPTION | COMP CODE | LENGTH   | ***** DATA CHECKS BY NUMBER OF RETRIES ***** |     |       |      |      |     |       |       |      |     |  |  |  |       |
|---------|---------------|-----|--------|--------|-----------|----------|--|-----|-------|------|------|-----|-------|-------|------|-----|--|--|--|-------|
|         |               |     |        |        |           |          | ONE  | TWO | THREE | FOUR | FIVE | SIX | SEVEN | EIGHT | NINE | TEN |  |  |  |       |
| FATS300 | TAPE1         | 383 | FAT002 | WRITE  | ABNORM    | 00116 FT |  |     |       |      |      |     |       |       |      |     |  |  |  | *PERM |
| FATS300 |               |     |        |        | PASS      | 1        | 4  | 3   | 1     | 0    | 3    | 0   | 0     | 2     | 0    |     |  |  |  | 10    |
| FATS300 | TAPE1         | 383 | FAT007 | WRITE  | NORMAL    | 00198 FT |  |     |       |      |      |     |       |       |      |     |  |  |  | *PERM |
| FATS300 |               |     |        |        | PASS      | 1        | 0  | 0   | 0     | 0    | 0    | 0   | 0     | 0     | 0    |     |  |  |  | 2     |
| FATS300 | TAPE1         | 383 | 000511 | WRITE  | NORMAL    | 02363 FT |  |     |       |      |      |     |       |       |      |     |  |  |  | *PERM |
| FATS300 |               |     |        |        | PASS      | 1        | 0  | 1   | 0     | 0    | 0    | 0   | 0     | 0     | 0    |     |  |  |  | 0     |
| FATS301 | END OF REPORT |     |        |        |           |          |  |     |       |      |      |     |       |       |      |     |  |  |  |       |

## 11.10 IDRC OVERVIEW

**DATA COMPACTION** The 3480XF subsystem feature called improved Data Recording Capability provides a highly transparent way to reduce tape volume requirements for data processing installations. This feature is also called IDRC, Data Compaction, or Compaction. Depending on the data involved, compaction can reduce the amount of data to be recorded by 50% or even more. Once the data is compacted it is combined with other logical blocks into physical blocks up to 65535 bytes in length. The combining of logical blocks causes a discrepancy between the logical number of blocks and block sizes and the physical blocks and block sizes actually occurring on compacted tape.

**USAGE** Compaction can be considered transparent since the application does not need to know about it. No application changes are necessary to take advantage of it. Indeed, not even the JCL has to be changed if an installation decides to make compaction the site default.

On the other hand, successful usage depends on the availability of IDRC capable hardware. A compacted tape cannot be used at another site which does not have the IDRC feature installed. **Before you use IDRC, insure that your backup site/disaster recovery site has installed the IDRC feature.**

**STANDARDS** Tape labeling standards for compacted tapes include the following:

- All header, trailer, and user labels are written in normal mode (i.e., without compaction).
- An indicator in the labels is set if the file is compacted.
- Every block of a compacted file is compacted.
- Every file of a compacted volume or volume set is compacted.
- Standard access methods will report any error if a compacted file is encountered on a device which is not capable. If the access method is not equipped to report the error, a hardware error will occur.

**FATS AND FATAR PROCESSING** FATS and FATAR in any version can read these tapes without a problem (FATAR can also write them). However, without knowledge of the compaction, earlier versions of FATS and FATAR will misrepresent the length of the tapes involved. This is due to the fact that FATS and FATAR assume the lengths and gap sizes of the logical blocks it processes.

FATS and FATAR, Version 4.2 and higher, compute the tape length from the statistics that are available from the hardware itself. Using the bytes and blocks written to or read from a tape instead of the logical bytes and blocks, FATS and FATAR compute the estimated length of tape processed. In general FATS and FATAR will report the logical bytes and blocks transferred as usual. In those places where IDRC physical statistics are reported, these figures are clearly labeled as IDRC values.

CONTINUED . . .

## 11.10 CONTINUED . . .

**FATAR** FATAR will recognize compaction when reading tape and report new statistics for the operation. At the end of each compacted file FATAR reports statistics such as the following:

FILE CONTAINED 14129 BLOCKS – 3534 IDRC (COMPACTED) BLOCKS

This means that the compacted tape contained 3534 physical (and compacted) blocks which were "de-compacted" into 14129 logical blocks by the IDRC hardware. FATAR does not "de-compact" the blocks, that is done by the hardware IDRC feature. FATS and FATAR do not "see" the compacted blocks at all. Only the hardware "sees" the compacted data.

When a compacted file has been found on a tape, FATAR will also report tape savings under the heading FINAL TOTALS in the following manner:

IDRC DATA COMPACTION REDUCED TAPE REQUIREMENTS BY 54%

This means that the preceding file or files would have used more than twice as much tape. For instance, if the preceding message occurred for a single reel with a single file which also showed 512 under "FEET READ", the file would have used 2+ reels if it was not compacted.

FATAR issues an additional IDRC line in the TAPE SUMMARY to report the physical bytes and blocks on tape. On this line you will see a second average block size which is different from the logical average blocksize above it. This value is computed from the blocks and bytes read or written as reported by the hardware.

**FATS** FATS will recognize compaction when reading (READ operation) and report the appropriate statistics. A new detail action message, message id FATS209, action "IDRC FILE SENSED", will be produced for every compacted file encountered during the READ function.

No compaction will be used for the WRITE, LABEL, or ERASE operations, regardless of JCL used or installation defaults. Compaction in these operations is meaningless. Labels are not to be compacted. Erased tape contains no data. Certification is used to detect tape media problems. Compaction would prevent accurate determination of the location and length of the problem tape surface area.

## SAMPLE PART OF REPORT WITH IDRC STATS

## SUMMARIZE TAPES

Summary Report Lists All Files on the Tapes

| TAPE SUMMARY FOR TAPE VOLUME -000511- AT DENSITY 6250 BPI |        |      |          |          |       |               |          |       |      |                  |      |      |      | 8/02/95 |      |
|---|--------|------|----------|----------|-------|---------------|----------|-------|------|------------------|------|------|------|---------|------|
| OWNER - INNOVATION  |        |      |          |          |       |               |          |       |      |                  |      |      |      |         |      |
| PHYS DATASET NAME   | FILE   | FIL# |          | REC-     | LRECL | CREATING      | BLOCKS   | BYTES | PERM | ---BLOCKSIZES--- |      |      | EST. |         |      |
| FILE (LAST 17 CHARS)                                      | SERIAL | VOL# | CRDATE   | EXPDATE  | FM    | BLKSZ         | JOB&STEP | SEC   | READ | READ             | TEMP | MIN  | AVG  | MAX     | FEET |
| 2 FAT66.TEST1   | 000511 | 0001 | 1985/214 | 00/000   | FB    | 00080         | FATMANEX | 0     | 100  | 800000           | 0    | 8000 | 8000 | 8000    | 14   |
|   |        | 0001 |          |          |       | 08000         | FAT66A   |       |      |                  | 0    |      |      |         |      |
| 5 FAT66.TEST2   | 000511 | 0002 | 1985/214 | 1985/254 | VB    | 00095         | FATMANEX | 0     | 193  | 1425772          | 0    | 2284 | 7387 | 7414    | 24   |
|   |        | 0001 |          |          |       | 07500         | FAT66B   |       |      |                  | 2    |      |      |         |      |
| 8 FAT66.TEST3   | 000511 | 0003 | 1985/214 | 1986/001 | FB    | 00500         | FATMANEX | 0     | 700  | 3500000          | 1    | 5000 | 5000 | 5000    | 65   |
|   |        | 0001 |          |          |       | 05000         | FAT66C   |       |      |                  | 0    |      |      |         |      |
|   |        |      |          | -----    |       |               |          | ----- |      |                  |      |      |      | ----    |      |
| HIGHEST EXPIRATION =====>                                 |        |      |          | 1986/001 |       | TOTALS =====> |          | 993   |      | 5725772          |      | 3    |      | 103     |      |

## 11.11 3490E OVERVIEW

**3490E MODE RECORDING** The 3490E tape subsystem provides further capability to reduce tape volume requirements for data processing installations. Doubling the capacity of the 3480 cartridges is done by interleaving tracks on the cartridge. When a 3490 drive reaches the end of the usable tape it signals "logical end-of-tape", but the 3490E drive will simply reverse and continue writing data in the opposite direction until the beginning of the tape is reached (when "logical EOT" is finally signaled). The tracks for the forward direction (called "WRAP 1") and the reverse direction ("WRAP 2") are actually interleaved.

The 3490E's double-density is achieved by its ability to write 36 tracks across the tape, but it does not actually write any more densely or any faster than the 3480s. The 3490E also implements "auto-blocking" for all files. This feature was implemented on the 3480XF only when compaction was turned on. It is always active on the 3490E. This contributes a small additional percentage to the reduction of tape volume requirements.

**USAGE** As with IDRC, the 3490E recording mode may be considered transparent since the application does not need to know about it. There may be some installation JCL changes necessary to effectively use the 3490E. It is a different device type. The proper device type must be used to use the 3490E devices. Since the 3490E is capable to read 3480 and 3480XF tapes some new esoteric names have been created to accomodate the users.

- UNIT=3490 - to allocate a 3490E drive
- UNIT=SYS3480R - to allocate any 3480/3490/3490E drive
- UNIT=SYS348XR - to allocate any 3480/3490/3490E drive capable of reading a 3480XF format tape.

Compatibility is a larger issue to be considered. The 3490E subsystem can read 3480 and 3480XF format cartridges but cannot write them. The narrower heads of the 3490E are able to read the wider 3480 tracks but cannot record them for subsequent use on a 3480. An installation which has only 3490E drives can no longer write a cartridge which can be read by another site with only 3480/3490 drives.

**Please note that tapes which were written on a 3480/3490 can be read by a 3490E, but cartridges written by a 3490E can be read ONLY by a 3490E; there is no option on a 3490E to write 3480 format files. Be sure that your disaster site and any site you exchange tapes with can read the new format.**

**FATS AND FATAR PROCESSING** Earlier releases of FATS/FATAR prior to 4.2 Level 1 cannot handle 3490E tape subsystems due to incompatibilities in the CCW instructions of the 3490E. Earlier releases of FATAR will process normally on 3490E tape subsystems but will not be able to detect the 3490E mode and will report erroneous statistics about tape length.

CONTINUED . . .

## 11.11 CONTINUED . . .

**FATAR** Several informational FATAR messages were changed to support 3490E format tapes. Since there is no explicit indication of compaction associated with a file or block another method is required to determine whether compaction is involved. A comparison will be made between channel bytes and tape bytes obtained by a READ BUFFERED LOG command. If a difference is detected this file will be considered compacted. At the end of each compacted file FATAR reports statistics such as the following:

FILE CONTAINED 27500 BLOCKS – 6876 IDRC (COMPACTED) BLOCKS

There will be an additional informational message in the detail report to indicate when physical end of tape was reached and 3490E wrap 2 mode initiated. This message will have similar information about blocks and, conditionally, IDRC blocks. The difference will be in the text which will read

WRAP 2 AFTER 14158 BLOCKS – 3540 IDRC (COMPACTED) BLOCKS

The boxed summary which appears under the FINAL TOTALS will now have three forms:

- IDRC DATA COMPACTION REDUCED TAPE REQUIREMENTS BY nn%
- 3490E AND COMPACTION REDUCED TAPE REQUIREMENTS BY nn%
- 3490E FORMATTING HAS REDUCED TAPE REQUIREMENTS BY nn% “,TRTCH=COMP” CAN FURTHER REDUCE TAPE REQUIREMENTS

The calculation for reduced tape requirements will be adjusted for the additional length available on all cartridges in 3490E format. Consequently a file which would have required (4) 3480 format cartridges and fits on a single 3490E format compacted cartridge will show savings of 75% or greater.

**FATS** The FATS209 message will have 4 action descriptions. They are:

3480 FORMAT FILE SENSED  
 3480XF FORMAT FILE SENSED  
 3490E FORMAT FILE SENSED  
 3490E WRAP 2 REVERSAL SENSED

The action description

3480XF FORMAT FILE SENSED

will be produced for every compacted file encountered during the read function.

The action description

3490E FORMAT FILE SENSED

merely indicated the file (and tape) is in the new 36-track bidirectional Enhanced Capability tape format and does not provide any indication of compaction or non-compaction of the data.

The message FATS206 will be added. The action description associated with the message will be:

DATA COMPACTED BY nn%

This message will be issued when we determine that tape bytes differ from channel bytes for any file. This is the only indication available to detect whether IDRC compaction was used for the 3490E format cartridge.

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## 12.0 FATS GUIDELINES

### USAGE GUIDELINES

Many customers have asked for suggestions on the use of FATS as part of a regular program of tape maintenance. The following section contains explanations and guidelines designed to help you in the development of such a program. However, each installation should develop its own cost-justified procedure establishing what level of certification is required, at what point tapes are to be discarded, how often archival tapes are to be verified, and so forth. These guidelines are presented only as an example and an aid.

This section also contains answers to questions on FATS which customers have often asked.

## 12.1 CAUSES OF TAPE ERRORS

### CAUSES OF TAPE ERRORS

To better understand the usage of FATS you must understand the causes of error on magnetic tape. Any of the following can cause errors:

1. Excessive oxide on the tape surface.
2. Dirt or debris on the surface (often microscopic).
3. An oxide coating on the tape surface which does not retain an acceptable signal.
4. Tape surface damage such as pinholes, scratches, creases, and edge damage.
5. Dirt or oxide buildup on the tape drive heads.
6. Malfunction of the tape drive itself.

Many errors may be caused by loose oxide or dirt on the tape surface. These errors may show up as temporary (meaning that FATS retry actions caused the contaminant to be dislodged), or permanent. Even if they are permanent, surface cleaning by a Stand-Alone tape cleaning machine, or repeated certifications by FATS, may cause the errors to disappear.

Excess oxide is common on new tapes, as the manufacturing process may leave debris on the surfaces when the original wide tape is cut into 1/2 inch strips. New tapes should be run through a tape cleaner, if available, once before being certified by FATS.

Truly permanent errors may be caused by defective oxide coatings or surface damage. These errors cannot be rectified by FATS or a tape cleaner.

Dirty tape heads or tape drive malfunctions may cause reported errors which do not truly exist on the tape itself. FATS depends on the proper functioning of the tape drive to detect and report errors, so any errors in the tape drive mechanism will produce spurious errors. We recommend that the tape drive heads be cleaned at least every 10 tapes certified (more often if certifying brand new tapes); the FATS parameters "MAXCLEAN=" and "ERRCLEAN=" can be used to insure this. If FATS reports many errors on a tape, we recommend re-certifying it on another tape drive before deciding to discard it.

## 12.2 OPERATING SYSTEM VS. FATS ERRORS

### OPERATING SYSTEM VS. FATS ERRORS

We are often asked why a tape on which FATS has reported one or more errors can be used successfully as an output tape by an application program. The answer has to do with the purpose of FATS error reporting versus the operating system error recovery ([full details of error recovery and FATS operation are found in the section "FATS Technical Summary"](#)).

### FATS PURPOSE

FATS certification has the purpose of identifying every section of the tape (in increments of a few inches) which has a permanent or temporary write error. As explained below, this can allow you to predict where the operating system will experience permanent write errors or degradation due to error recovery.

CONTINUED . . .

## 12.0 CONTINUED . . .

### OPERATING SYSTEM ERROR RECOVERY

Operating system error recovery has the purpose of letting the application program run to successful completion despite I/O errors on tape. To accomplish this, write errors are retried by erasing a length of tape and attempting to write the block again. Depending on the location of the error, and the length of the block, this action may need to be repeated until it is successful or until error recovery considers it a permanent error (16 attempts) at which point the application program is usually terminated.

As an example, assume that FATS has identified a permanent error in the middle of a tape with no other errors around it. That tape is being used by an application program writing blocks of length 16000 bytes at 1600 BPI (10 inches long). At the first attempt, that permanent error falls in the last inch of the block. The operating system erases 3.45 inches of tape and writes again, but the error is still within the block (now about 5.5 inches into it). The system erases 3.45 inches again, and the error is now about 2 inches from the start of the block. 3.45 inches are erased again, and the block is now written past the error and writes successfully. No error is apparent to the program or the operator, but time was wasted doing this recovery, and almost a foot of blank tape was skipped over.

If however, there is a cluster of permanent errors on the tape, spaced 6 or 8 inches apart, this same situation will probably result in a permanent write error indication from error recovery. As error recovery erases tape (3.45 inches at a time) to move the block up, it keeps encountering a new error on the tape, since the 10 inch data block is longer than the distance between errors. Obviously, a given pattern of permanent errors reported by FATS may or may not result in a permanent error to an application program depending on the tape density and data block length.

### TAPE ERROR STATISTICS

Although tape write errors may be bypassed by this error recovery process, the existence of the errors is always recorded by the operating system. The system error recording file "SYS1.LOGREC" will always contain the total number of temporary and permanent read and write errors encountered on every tape volume processed, and can be printed by the EREP (Environmental Recording, Editing, and Printing) program appropriate for your system. If the SMF sub-system is set up to record type 21 ESV (Error Statistics by Volume), these records will also contain this information and can be printed by the IFHSTATR utility documented in the appropriate system utilities manual. These statistics may help to make clear the hidden cost of "unseen" tape errors.

## 12.3 VARIATIONS IN FATS RESULTS

We are often asked why the results of a FATS certification or verification will vary on two consecutive runs on the same tape. If you review the causes of tape errors above, the reasons are fairly obvious.

For errors caused by dirt or debris on the tape, these contaminants may be removed by one FATS execution, or may be picked up by the tape drive heads and re-deposited in another location, causing an error to "move", even if re-certified on the same drive.

If errors are caused by marginal oxide surface on the tape, the surface may be able to record and reread a signal using the amplifiers and heads on one tape drive, while being just barely inadequate on another, causing errors to appear and disappear.

Naturally, if the errors are caused by dirty tape heads or tape drive malfunction, the results will be different on different drives.

## 12.4 ANALYSIS OF CERTIFICATION RESULTS

FATS certification ("WRITE" function) is designed for new tapes or tapes currently in scratch status in order to decide if they are acceptable for use as output tapes. It is impossible to make firm recommendations on the action to take on a particular tape based on its FATS results; each installation must decide for itself what level of error recovery degradation it is willing to accept (based on the number of permanent errors on a tape) versus the cost of replacing a tape volume. However, we can provide some guidelines for arriving at this value, as well as some circumstances when action is definitely indicated.

These decisions will usually require examination of the FATS detail report to analyze the number and position of the errors on a given tape. It is possible to automate some of this decision process (the FATS parameters "MAXERR=" and "MAXCERR=" will cause automatic rejection of a tape with an excessive number of errors or contiguous errors), but most of the analysis is beyond the capabilities of FATS to perform dynamically. The user will have to examine the total number of permanent errors, the location on the tape of these errors, and the clustering of the errors (FATS will report on errors which are contiguous, but some errors may be separated by small lengths of good tape but still be close enough to cause problems).

In most installations, the majority of tape datasets occupy only the first 10-20% of the tape volume. As a result, the beginning of the tape gets the most repeated use and is subject to the most errors due to tape wear. If a tape shows permanent errors (even only 1 or 2) in the early part of the tape, some action is indicated. It is possible to cut off the portion of the tape containing the errors (except tape cartridges) and put a new beginning-of-tape reflective marker on it, resulting in a shorter but usable tape. Most installations set some minimum acceptable length for a tape (such as 2000 or 2200 feet); since FATS shows both the total length of the tape and the length to remove (the location of the last error), you can easily calculate whether a tape is worth this "stripping" action.

Next, look for contiguous or clustered permanent errors. Clusters of errors exceeding 4 inches in length on IBM 3480/3490 cartridges or 3 feet in length on other tapes are almost certain to cause permanent write errors; clusters shorter than this may be recoverable but will cause degradation. If a tape contains any long clusters or more than one short cluster, it should be removed from use.

If a tape is free from the above problems, it can be categorized by the number of permanent errors. The following categories are suggestions only:

For Round Tape:

0-10 Errors – Good tape

11-25 Errors – Marginal tape. Clean if possible. Do not use for critical applications.

Over 25 Errors – Bad tape. If possible, clean and re-certify. If errors persist, remove from use.

For Cartridges:

0-5 Errors – Good tape

Over 5 Errors – Clean and re-certify. If errors persist, remove from use.

In summary, the recommended analysis procedure is:

- 1) If the tape contains errors only or mainly in the beginning, strip the defective tape if its remaining length is acceptable.
- 2) If the tape has large or numerous clustered errors, discard it.
- 3) Make a decision based on the total permanent errors.

## 12.5 ANALYSIS OF VERIFICATION RESULTS

**VERIFICATION RESULTS** FATS verification ("READ" or "ANALYZE" functions) have the purpose of verifying the readability of existing data files on tape. This can be used for various critical production data (to avoid data checks in the middle of critical runs) or verification of archival data which is rarely read but which must be retained.

FATS will report on every data block (by file number and block number within file) on which a permanent or temporary data check occurred. FATS does not attempt to reproduce the complete actions of the operating system error recovery in the case of a read data check, so it is possible that a data check reported by FATS will in fact be readable by an application program.

However, even one data check on such a tape means that the tape is suspect and may cause application failures now or in the future. To be safe, the tape should be copied as soon as possible to a certified scratch tape. Because of its superior facilities for recovering from data checks and its ability to copy multiple files, the FATAR program is recommended for this purpose.

## 12.6 RECOMMENDATIONS FOR FATS USAGE

The following is a suggestion for a formal program of FATS usage to certify scratch tapes and verify data files. As explained above, each installation will have to select the intervals and values that make sense for it.

**ARCHIVAL DATA TAPES** Archival data tapes (long term retention tapes which are rarely read but must be retained for legal or other reasons) should be verified for readability at intervals not exceeding one year. Tapes which sit on the rack for long periods have a tendency to "dry out" causing data checks. The action of reading them with FATS will help to prevent this as well as identifying any data checks which have occurred. Any data check is good reason to copy the tape.

**CRITICAL DATA TAPES** It is reasonable to assume that a tape which was recently written successfully will be readable without error, but if certain data tapes are critical to the successful execution of important programs, especially if those programs are long-running or difficult to restart, it is useful to run FATS against each data file before starting the application. This verification only takes a few minutes per tape, but may save hours of recovery if data checks are found.

**NEW TAPES** New tapes should be cleaned (if a manual cleaner is available), and certified and labeled using FATS. If the tapes cannot be separately cleaned and they do get errors, certify them several times to remove all loose oxide; be sure to clean the tape drives frequently. If certification of a new tape discovers errors which would classify it as a marginal or bad tape, you may wish to discuss it with your tape supplier.

**SCRATCH TAPES** Tapes which are in scratch status (available for use as output tapes), should be certified if:

- a) A certain interval (such as 4 months) has elapsed since their last cleaning
- b) They have been used for output more than a certain number of times (such as 50)
- c) They have received more than a certain number of temporary or permanent read or write errors (such as 5) as reported by SMF or EREP.

Some tape management systems will track these numbers and recommend cleaning/certification of specific volumes based on user provided thresholds. Disposition of the certified tape should be based on the recommended analysis technique outlined above.

## 12.7 QUESTIONS AND ANSWERS

These are some commonly asked questions about FATS.

### **PASS=2 What are the benefits of the PASS=2 option?**

FATS certifies by writing data blocks on tape and noting which get write data checks. However, the normal function of the tape drive erases a short inter-block gap between each block; this gap is not checked by FATS. PASS=2 rewinds the tape after certification and re-certifies it, selecting a different starting location so that those gaps will be certified. This obviously takes twice as long as PASS=1. The normal operation of FATS will certify 90-97% of a tape's surface (depending on the tape density) in one pass; this is usually adequate to make a decision on the condition and usability of a tape. If 100% surface certification is a necessity, PASS=2 can be used.

### **DATA CHECKS AT EOVS Why do tapes sometimes show data checks at end of reel?**

The end of a tape reel (except on IBM 3480 cartridges) is indicated by a reflective marker on the tape. If the tape is wound too tightly by the manufacturer or by the tape drive, impressions of this marker will appear every 14 inches for several layers of tape above and below the reflector. These impressions can cause data checks, especially when writing large data blocks. FATS will report these errors as data checks about 1 foot apart at the end of the tape. The end of tape reflector can be moved up to eliminate this error.

### **6250 BPI Why do tapes which fail to record data at 1600 BPI sometimes record successfully at 6250 BPI?**

It would seem obvious that increased density would require a higher quality tape, and that a tape showing errors at 1600 BPI would show as many or more errors at 6250 BPI. However, this turns out not to be true, mainly due to technical improvements between the hardware used for 1600 and 6250 BPI. Briefly, they are:

- a) An improved tape cleaning mechanism
- b) Automatic Read Amplification (ARA) which automatically adjust the amplifiers to the characteristics of the tape.
- c) Improvements in the READ/WRITE circuitry
- d) Recording of RESYNC bursts in the data improving recovery from lost data bits.

These improvements make recovery from Read and Write errors much more likely, but the use of marginal tapes at 6250 BPI may mean that a great deal of operating system error recovery is taking place with its associated degradation. Even though permanent data checks may be less likely with this hardware, the use of FATS to improve the quality of the tape library can provide overall efficiency and reliability.

### **FATS RESOURCES What resources does FATS use?**

FATS CPU usage is minimal, but, by the nature of the function it performs, it is a heavy user of the tape channels and control units. We recommend that FATS be run primarily when other tape usage is low. If multiple FATS operations are being performed, FATS will perform better if the tape drives are on separate channels and control units, if possible. There is little point in allocating more than three tape drives on the same channel or control unit to FATS, since no performance gain will result. Note that this does not apply to the FATS "ERASE" function which does not tie up the channel or control unit. Also, 3480 tape cartridge sub-systems, because of their multiple channel paths and buffering, may be able to achieve better performance.

### **SMF/EREP RECORDING Why aren't FATS data checks reported in SMF/EREP?**

Temporary and permanent data checks are not reported in SMF (type 21 ESV records) and "SYS1.LOGREC" (EREP) unless the operating system error recovery was invoked for each error. FATS is able to do what it does only by inhibiting system error recovery, so these errors will not be reported. An exception is the 3480 tape cartridges sub-system, where the control unit accumulates the errors and reports them to SMF/EREP; however, every FATS retry will count as one error, so the SMF/EREP values will be much higher than the FATS totals.

CONTINUED . . .

**12.7 CONTINUED . . .****RETRY LEVEL What retry level should be used?**

Although the default value for "RETRY=" in FATS is 10, experience and analysis of many FATS reports has shown that if an error repeats 5 times, it is likely to still be present after 10 retries. If a tape has many errors, specifying "RETRY=5" may significantly reduce the elapsed time of the certification, while still providing the same information on the condition of the tape.

**FULL-TAPE Why can't FATS READ verify to the physical end-of-tape?****VERIFY**

Many customers have asked for the ability in FATS to READ verify to the physical end-of-tape reflector, rather than stopping at two tape marks (logical end-of-data). This is not practical for several reasons:

- a) The tape drive raises an exception condition when you attempt to write past the end-of-tape (this is how the FATS WRITE operation knows when to stop), but it gives no indication when you are reading past the end-of-tape reflector.
- b) It is possible to discover if the tape is positioned past the reflector (at the cost of an extra I/O operation for every data block), but this will work only if there are data blocks recorded out to the end-of-tape. If there is blank tape preceding the reflector, the first read in this blank area will not end until the tape runs off the reel; FATS has no way to predict and avoid this.

In any case, the function of FATS verification is to prove the readability of the recorded data on the tape. There is no point in verifying data past the current logical end-of-data.

**12.8 CERTIFICATION OF TAPE CARTRIDGES**

When IBM released the 3480 cartridge sub-system in 1985, the error recovery within the tape controller and the higher quality of the media resulted in a significant reduction in the number of data checks compared to 6250 BPI tape. As with any media, with continual use, the number of errors will tend to rise as the average age of your tape library rises.

FATS gives you the tool to certify cartridges and insure that cartridges that have been in the library for a number of years and have potential for errors be identified and eliminated from the tape library.

With the 3490E drive increasing the number of tracks from 18 to 36 without changing the media, the number of tape errors can increase.

While the overall tape quality from manufacturers has been good, a number of manufacturers have had at times quality control problems. While the manufacturers have replaced the tapes, the data center has had to invest a significant amount of time and CPU resources to copy all of the historical data from the bad tapes to newer tapes. FATS, if used as a regular tool for certification can help to identify these conditions at an early stage.

## 13.0 FATS EXECUTION JCL

To execute FATS, the following JCL statements are required:

|  |   |
|--|---|
| <b>JOB STATEMENT</b>                         | User supplied.  |
| <b>STEPLIB/<br/>JOB LIB DD<br/>STATEMENT</b> | A STEPLIB or JOBLIB DD Statement will be required if FATS has been linked into a private library. It can be omitted if FATS is in a system library which can be accessed without a STEPLIB/ JOBLIB statement (that is, a library in the system link list such as "SYS1.LINKLIB").   |
| <b>EXEC STATEMENT</b>                        | <p>The EXEC statement specifies the FATS program name, memory requirements (if your installation defaults are insufficient), and parameter field.</p> <p>The basic memory requirements of FATS are 256K. If you use the "BLKSIZE=" parameter to increase the size of the block written by FATS above 32K, this value must be added to the basic requirement. If FATS' companion product FATAR is invoked as a FATS sub-task (via the "ANALYZE(n)" control statement), then the memory requirements of each concurrent FATAR sub-task must be added to the basic requirement (FATAR requirements are documented in another section of this manual).</p> <p>If the EXEC statement parameter field contains an "I" ("PARM=I"), FATS will print on SYSPRINT a brief summary of its control statements and messages.</p> |
| <b>SYSPRINT DD STATEMENT</b>                 | SYSPRINT receives the listing of all FATS control statements, a summary of the operations to be performed, and messages about major errors. In addition, it may receive the detail reports for each tape drive and the summary report if the DD Statements for those reports are absent. SYSPRINT is normally allocated to a SYSOUT data set. Its DCB attributes are RECFM=FBA,LRECL=121. If blocksize is specified, it must be a multiple of 121, otherwise it will default to 121 for SYSOUT or 1210 for other devices.   |
| <b>SYSPRINx DD STATEMENT</b>                 | SYSPRINx receives the detail report for the operations performed against tapes mounted on DD Statement "TAPEx", where "x" is a digit from 1 to 9. This allows the detail reports for simultaneous FATS operations to be printed separately. SYSPRINx is normally allocated to a SYSOUT data set. If SYSPRINx is not present for a particular TAPEx, its detail report will go to SYSPRINT. DCB attributes for SYSPRINx are the same as those for SYSPRINT.  |
| <b>SUMMPRT DD STATEMENT</b>                  | SUMMPRT receives the summary report, consisting of several lines summarizing the results of FATS operation against each tape processed. SUMMPRT is normally allocated to a SYSOUT data set. If SUMMPRT is not present, summary reports will go to the "SYSPRINx" data set associated with the "TAPEx" DD Statement on which the tape was mounted; if it is also absent, summary reports go to SYSPRINT. DCB attributes are the same as for SYSPRINT.  |
| <b>SYSUDUMP DD STATEMENT</b>                 | SYSUDUMP requests an abend dump if major errors occur (note that most internal abends in FATS are for the user's information only and do not cause dumps). SYSUDUMP is usually allocated to SYSOUT. Abend dumps are necessary for analysis of problems by INNOVATION DATA PROCESSING.   |

CONTINUED . . .

## 13.0 CONTINUED . . .

**TAPE<sub>x</sub> DD  
STATEMENT**

TAPE<sub>x</sub> defines a tape unit (and optionally a volume) to be used for one operation by FATS. "x" is a digit from 1 to 9 and corresponds to the digit on a FATS control statement (e.g., "WRITE(1)" uses DD statement "TAPE1").

If used for output (FATS operations "WRITE", "LABEL" or "ERASE"), the DD statement should say "DISP=(NEW,KEEP)". A volume serial (VOL=SER=) can be specified to request a specific volume, or omitted to request a scratch volume. The volume serial can also be omitted when FATS is supplying the serials internally (via its own "VOL=" parameter).

**NOTE:** if a volume serial is specified, the label type is "SL", and the FATS "MULT" parameter is given, the volume serial will be blanked to request scratch tapes for each mount after the first.

If used for input (FATS operations "READ" and "ANALYZE"), "DISP=(OLD,KEEP)" should be specified, and a volume serial must be given (if the FATS parameter "VOL=" is being used to supply serial numbers, the volume serial in JCL may be any dummy serial).

If FATS parameter "VOL=" is used to supply volume serials, the JCL parameter "DEFER" (as in "UNIT=(xxxx,DEFER)") should be used so that the operating system will not call for a tape mount before FATS fills in the proper volume serial.

Any type of label processing may be specified, but it is frequently desirable with FATS to specify "LABEL=(,BLP)". If installation conventions do not allow the use of "BLP", the FATS parameter "BLP" can be used to internally change the label type to "BLP" before open; "BLP" can be used under MVS only if FATS has been installed as an authorized program, and installation open exits may still prevent its use.

**WARNING: For proper operation of FATS, the TAPE<sub>x</sub> DD statement must contain a DSN= parameter and it must not specify a temporary (&&) name. Also the parameters DISP=(...,PASS) or VOL=(...,RETAIN) must not be specified. Otherwise, the tape may run off the reel or other errors may occur.**

**TAPE<sub>x</sub>OUT DD  
STATEMENT**

TAPE<sub>x</sub>OUT (where "x" is the same 1 digit number as in "TAPE<sub>x</sub>" above) is used only for the FATS "ANALYZE(x)" function, which causes the companion product FATAR to be invoked. If present, TAPE<sub>x</sub>OUT will request FATAR to copy the tape mounted on TAPE<sub>x</sub>. See FATAR documentation elsewhere in this manual for details.

**SYSIN DD  
STATEMENT**

The SYSIN DD Statement is the source of FATS control statements. It is normally a "DD \*" spool file, but can be any disk or tape file with DCB characteristics RECFM=FB and LRECL=80.

### 13.1 OPERATOR CONTROL

- OPERATOR CONTROL** By default (WTOR=YES parameter), FATS issues a WTOR to the operator's console at startup. This outstanding reply may be used to control FATS processing. The standard OS operator commands STOP (P) and MODIFY (F) may also be used to control FATS processing.
- P id** This command will cause FATS to halt all tapes in progress, cancel any pending mounts, and terminate with a U0888 abend or return code 12 if the optional parameter MODIFY=YES was specified in the FATS control cards (default). This is similar in operation to replying KEOJ to the outstanding WTOR.
- F id,QUIT** This command will terminate FATS after all tapes which are currently in progress have completed if the optional parameter MODIFY=YES was specified in the FATS control cards (default). Any pending mounts must be satisfied. It is similar to the EOJ reply to the outstanding WTOR.
- F id,KUUU** This command will stop processing on the specified tape unit and, if appropriate, may request another tape be mounted on the drive to continue processing if the optional parameter MODIFY=YES was specified in the FATS control cards (default). It is similar to replying KUUU to the outstanding WTOR.
- id** The parameter "id" in the commands above is the jobname of the FATS job if it was initiated by standard job submission. If the operator started FATS by issuing a start command, "id" is the name of the procedure started or the id given on the start command.

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## 14.0 FATS CONTROL STATEMENTS

### 14.1 GENERAL

**GENERAL RULES** All FATS control statements must be input on 80-character records of which only columns 1 to 71 are used by FATS. Each control statement contains:

**OPERATION KEYWORD** Identifies the operation to be performed by FATS. It can start in any column from 1 to 50 on the input record, but any columns before it must be blank. Only one operation keyword can be specified for each tape drive used by FATS. Valid operation keywords are:

- DEFAULT – Set defaults for other operations
- WRITE – Certify a tape
- READ – Verify readability of a tape
- ERASE – Erase data from a tape
- LABEL – Write standard labels on a tape
- ANALYZE – Invoke FATAR to read/copy a tape

**TAPE NUMBER** Immediately follows the operation keyword (no intervening blank spaces) and is enclosed in parentheses. It must be a single digit from 1 to 9 and identifies to which TAPEX DD Statement this operation applies (e.g., "(1)" indicates DD Statement TAPE1). Each digit may appear on only one control statement in a given FATS execution. The tape number is not used on the DEFAULT control statement, but is required on all other control statements.

**OPTION PARAMETERS** Are separated from the operation keyword (and tape number) by one or more blank spaces. Each control statement may have one or more option parameters (separated by commas) or none at all. The options modify the function of the operation keyword; they control labeling, tape mode, error retries, etc. The last option parameter must be followed by a blank space.

**CONTINUATION** Options can be continued onto another control statement by following the last parameter with a comma and a blank space, and continuing the parameters in any column of the next statement.

**COMMENTS** May appear on any FATS control statement. They must be separated from the last option parameter by at least one blank space (a control statement containing no options cannot contain comments). Any control statement with an asterisk ("\*") in column 1 will be bypassed by FATS and treated entirely as comments.

## 14.2 OPERATION KEYWORDS

- DEFAULT** Used to change the defaults of FATS option parameters. It affects any other operation keywords which follow it. If several operations are to be used in a FATS step, and they all require the same option specifications, DEFAULT is a simple way to enter the options only once.
- WRITE(n)** Used to certify the quality of new or old scratch tapes by writing special patterns on the entire length of the tape and checking for write errors. This will destroy any existing data on the tape, so the tape must be in scratch status.
- READ(n)** Used to verify the readability of existing data on a tape. READ(n) can read one or more data files, but cannot read past the end of currently recorded data since FATS has no way of detecting the end-of-tape marker in read mode. Can be used on non-scratch tapes since it will not destroy any data.
- ERASE(n)** Invokes the "Data Security Erase" hardware function to erase all data from a tape. It may be used to totally erase a tape, or to erase data past currently existing files. On 3480 cartridge drives, this involves writing a random pattern for added security. Unlike "WRITE(n)", the "Data Security Erase" does not involve the tape control unit, so other FATS operations or tape I/O from other jobs can proceed without contention.
- LABEL(n)** Writes an IBM standard volume label and data set labels on a scratch tape. LABEL(n) can be used to label new tapes or to re-initialize old tapes. It does destroy data on the tape.
- ANALYZE(n)** Invokes FATS' companion product FATAR to read and optionally copy a tape. Complete information on FATAR is contained in other sections of this manual. Although FATAR can be executed as an independent program, execution under FATS allows multiple FATAR tasks to execute in one jobstep and allows the use of FATS parameters for the selection of tapes to be processed. Your installation must be licensed for FATAR to use this operation.

### 14.3 OPTION PARAMETERS

#### OPTION PARAMETERS

All option parameters can appear on all operation statements, but some actually apply only to one or more of the operation keywords. They are grouped together by their functional type (as shown in the section headings below), but parameters which apply only to one or more operations are marked as such. All option parameters (except "VOL=") can appear on a DEFAULT Statement.

#### 14.3.1 RETRY/ERROR PARAMETERS

- ERRCLEAN=n** Applies to: WRITE, READ  
If more than "n" permanent or temporary errors are encountered on a tape, FATS will request the operator to clean the tape drive, and then re-certify or re-verify that tape. "n" may be from 1 to 32767. This cleaning action will be requested only once per tape drive.  
Default: 32767
- MAXCERR=n** Applies to: WRITE, READ  
If contiguous permanent errors totaling more than "n" inches are encountered on a tape, FATS will terminate processing of that tape. "n" may be from 1 to 32767.  
Default: 200
- MAXERR=n** Applies to: WRITE, READ, ANALYZE  
If more than "n" permanent errors are encountered on a tape, FATS will stop processing of that tape. "n" may be from 1 to 32767.  
Default: 200
- PASS=1** Applies to: WRITE  
**PASS=2** This option specifies the number of times that FATS will write over a tape during certification. With "PASS=1" (the default), FATS will certify the entire surface of the tape except for the inter-record gaps between the pattern records written by FATS, resulting in 90-97% certification depending on tape density. "PASS=2" will cause the tape to be rewound and recertified using a different starting location on the tape which will insure 100% certification. Some errors reported during the first pass may appear again during the second pass.  
Default: PASS=1
- RETRY=h** Applies to: WRITE, READ, ANALYZE  
**RETRY=(l,h)** "h" defines number of retries required before an error is considered permanent, and "l" the number of retries before an error (temporary or permanent) will be printed in the detail report. "h" may have values from 1 to 99, and "l" may go from 0 to 99 but must be less than "h". The value of "h" affects the summary report: for 10 or less the summary report will show temporary and permanent errors by number of retries; for over 10 all temporary and permanent errors will be totaled.  
Default: (1,10)
- STOP** Applies to: WRITE, READ  
When this option is specified, FATS will stop and issue message FATSW05 to the system operator whenever a permanent error is encountered, allowing inspection of the tape. After inspection, the operator may reply to the message requesting continuation (and stopping at the next error), termination, or continuation without stopping.  
Default: Do not stop on errors.
- STOPNUM=n** Applies to: WRITE,READ  
Invokes the "STOP" option (above) after "n" permanent errors have been encountered, allowing the first "n" errors to be skipped without operator interaction. Do not specify "STOP" if "STOPNUM=n" is specified.  
Default: Do not stop on errors.

## 14.3 CONTINUED . . .

## 14.3.2 VOLUME MOUNTING PARAMETERS

**BLP** Applies to: ALL  
 The "BLP" parameter is provided to help when installation conventions do not allow "LABEL=(,BLP)" to be specified on a JCL DD statement, but it desired to bypass label processing on FATS tapes. This is especially useful when certifying/labeling virgin or unknown tapes, or when reading multiple input tapes with varying data set names. When "BLP" is specified, the JCL label parameter is changed to "BLP" before the tape is opened. "BLP" can be used under MVS systems only if FATS is installed and executed as an authorized program (see the installation instructions). "BLP" may be disallowed by installation open exits.  
 Default: Honor the JCL "LABEL" parameter.

**MAXCLEAN=n** Applies to: ALL  
 When multiple tapes are being processed on a tape drive, specifies that the system operator is to be requested (via console message FATSW06) to clean the tape drive after every "n" tapes are completed. This is to prevent dirt or oxide deposited on the read/write heads by previous tapes from causing spurious errors.  
 Default: No cleaning requested between tapes.

**MAXVOLN=n** Applies to: ALL  
 Limits the number of volumes which FATS will request on a given tape drive when either the "MULT" or "VOLINCR=n" parameters are used.  
 Default: No limit for "MULT" or 682 for "VOLINCR=n".

**MULT** Applies to: ALL  
**NOMULT** "MULT" requests that multiple tapes be processed. As each tape is completed, a mount for a new tape will be issued. This will continue until the operator terminates FATS by replying to the FATS console message FATSW02, or until the number of tapes specified by the "MAXVOLN=n" parameter have been processed. "NOMULT" (the default) is used to turn off "MULT" if it has been specified on a DEFAULT statement.  
 Default: Process only one tape unless "VOL=" specified.

**VALIDATE=INPUT|OUTPUT|ALL** Applies to: WRITE, ERASE, READ, ANALYZE  
 If "LABEL=(,BLP)" is specified on TAPEX DD statements, data management will not verify the volume serial or expiration date of the tape. However, if "VALIDATE=" is specified, FATS will verify the volume serials of the mounted tapes for input tapes ("INPUT"), output tapes ("OUTPUT"), or both ("ALL"). The volume serial compared is the serial in the TAPEX DD statement or that provided by the FATS "VOL=" parameter. For output tapes, the expiration date in the tape labels will also be checked. If the wrong volume serial is mounted, or if the expiration date is not yet reached, FATS will issue message FATSW08 to the system operator giving the option of ignoring the error, mounting another tape, or skipping the tape. For WRITE and ERASE operations, validation will be done only if the "SAVLAB" option is specified or defaulted. VALIDATE= will be ignored if the "NOREWIND" option is given.  
 Default: No validation.

CONTINUED . . .

## 14.3 CONTINUED . . .

**VOL=v**  
**VOL=(v,...,v)**

Applies to: ALL (cannot appear on a DEFAULT statement).  
Specifies one or more tape volume serials to be processed by the operation statement on which it appears. "v" is a 1-to-6 character volume serial. "VOL=v" requests one volume. "VOL=(v,v,...,v)" requests multiple volumes; in this format, up to 682 serials may be supplied; the serial list may be continued onto multiple input statements by placing a blank space after any comma in the volume list, and continuing the list in any column of the next input record. When used in conjunction with the "VOLINCR=n" and "MAXVOLN=n" parameters, "VOL=" can be used to request a sequence of tapes without actually entering all of their serials. Note that "VOL=" on a "LABEL" operation implies that FATS will label the tapes with the given volume serial; on a "WRITE" or "ERASE" operation, this is true only if the "NOSAVE" parameter is also specified.

**VOLINCR=n**

Applies to: ALL  
This parameter can be used only in conjunction with the "VOL=" parameter. When specified, the last (or only) volume serial specified by "VOL=" will be incremented by "n" which must be a 1-to-6 digit decimal number. That volume serial must end in at least one numeric digit; only the trailing numeric part of the serial is incremented. The serial will be repeatedly incremented by "n" until either 682 volume serials have been generated or until the serial cannot be incremented further without "overflowing" the numeric portion. Although a large number of such serials may be generated by "VOLINCR=", the "MAXVOLN=n" parameter may be used to limit the number of volumes which will actually be requested by FATS. "VOLINCR=1" may be used to cause a number of consecutively numbered tapes to be requested.  
Default: No incremented serials are generated.

### 14.3.3 LABELING PARAMETERS

|                                |  |
|--------------------------------|--|
| <b>DD=LABEL</b>                | Applies to: WRITE, LABEL, ERASE<br>Causes the tape to be labeled with the volume serial specified on the associated TAPEX DD statement. Cannot be used with the "MULT" parameter.  |
| <b>LABEL=vvvvvv</b>            | Applies to: WRITE, LABEL, ERASE<br>Causes the tape to be labeled with volume serial "vvvvvv". Cannot be used with the "MULT" parameter.  |
| <b>NOLABEL</b>                 | Applies to: WRITE, ERASE, LABEL<br>Resets all LABEL options (including "SAVLAB"). The tape being processed will become an unlabeled tape. When used with the LABEL(n) operation, NOLABEL will quickly initialize unlabeled tapes.  |
| <b>OPERATOR<br/>NOOPERATOR</b> | Applies to: WRITE, LABEL, ERASE<br>If "OPERATOR" is specified, as each tape is opened, the system operator will receive message FATSW01 and must reply with the volume serial to be written on the tape. "NOOPERATOR" is used only in conjunction with "SAVLAB" (specified or by default) to prevent "SAVLAB" from going into "OPERATOR" mode if a tape is not labeled (see below), so that unlabeled tapes will remain unlabeled.   |
| <b>OWNERID='..'</b>            | Applies to: WRITE, LABEL, ERASE<br>Specifies the contents of the owner ID field in the volume label written by FATS. 1 to 10 characters (including blanks) may appear between the apostrophes. "OWNERID=" has an effect only if FATS is writing a volume label due to other labeling options. Even if "SAVLAB" is used, "OWNERID=" may be used to change the owner ID.<br>Default: 10 blank spaces (or original value if "SAVLAB")   |
| <b>SAVLAB<br/>NOSAVE</b>       | Applies to: WRITE, ERASE<br>If "SAVLAB" is specified or defaulted, and the tape being certified contains IBM standard labels, the tape will be re-labeled with its original volume serial; if the tape does not contain IBM standard labels, the tape will be labeled with a serial provided by the system operator (see "OPERATOR" above) unless the "NOOPERATOR" parameter is present. "NOSAVE" will prevent the volume serial from being preserved. Note that SAVLAB preserves only the volume serial and owner ID from the volume label; the contents of data set header labels is not saved.<br>Default: SAVLAB |

**14.3.4 PRINT PARAMETERS**

- LINECNT=N** Applies to: ALL  
Specifies the number of lines per page to be printed on FATS reports. If specified on a "DEFAULT" statement, it will control the page size of the FATS control report and summary report as well as providing a default for the detail reports. If specified on any other operation statement, it modifies only the detail report page size for that operation.  
Default: 56 lines per page
- MAXDETCNT=N** Applies to: ALL (except ANALYZE)  
Specifies the maximum number of detail report lines which will be printed together before double spacing. This will be effective only if a separate detail report DD statement "SYSPRINx" has been provided.  
Default: 4
- NEWPAGE** Applies to: ALL (except ANALYZE)  
When multiple tapes are processed on a tape drive, the "NEWPAGE" parameter requests that the detail report will skip to a new page whenever a new volume is processed, making the report easier to separate by volumes. This will be effective only if a separate detail report DD statement "SYSPRINx" has been provided.  
Default: Skip a single blank line between volumes.
- NONMETRIC  
METRIC** Applies to: ALL  
When NONMETRIC is specified (or defaulted) FATS will list tape lengths and error positions and lengths in American units. When METRIC is specified FATS will list tape lengths and error positions and lengths in metric units.  
Default: NONMETRIC.
- THRESHOLD=n** Applies to: WRITE, READ  
Specifies the maximum number of permanent and temporary errors which will be printed in the detail report for any given tape before printing is suspended. The summary report will still recap the total number of errors. This is used to limit the size of the printout.  
Default: 1000 errors.

**14.3.5 CONTROL PARAMETERS**

- MODIFY=YES|NO** Applies to: DEFAULT  
Specifies whether FATS will respond to an operator's MODIFY (F) and STOP (P) console commands. FATS ordinarily issues a WTOR to the console at startup which the operator can reply to at any time to control FATS execution (see the WTOR parameter). If the parameter MODIFY=YES is specified the operator may use the OS MODIFY (F) and STOP (P) commands to control FATS processing.  
Default: MODIFY=NO
- WTOR=YES|NO** Applies to: DEFAULT  
Specifies whether FATS will issue its WTOR message in order to provide operator control over FATS processing. With the MODIFY=YES parameter set to allow operator control with OS commands, it is not necessary to have the outstanding reply available to control processing.  
Default: WTOR=YES

## 14.3.6 MISCELLANEOUS PARAMETERS

|                                  |  |
|----------------------------------|--|
| <b>BLKSIZE=n</b>                 | <p>Applies to: WRITE</p> <p>Specifies the size of the block which will be written to tape during certification, in bytes. "n" must be less than 65536 and more than twice the density (BPI) of the tape being certified (except for 3480 tape cartridges).</p> <p>Default: Varies by tape density (i.e., 32760 at 6250 BPI).</p>   |
| <b>BPI=bbb</b><br><b>GAP=ggg</b> | <p>Applies to: WRITE, READ</p> <p>These parameters are used by FATS to support non-standard tape drives, drives which are not manufactured by IBM and are not compatible with an IBM drive, e.g., a 3200 BPI drive. To use such a drive, specify "MODE=03", "BPI=bbb", and "GAP=ggg" where "bbb" is the density of the drive (in bytes per inch) and "ggg" is the byte-equivalent of the inter-record gap size of the drive (calculated as the gap size in inches times the BPI). With these parameters specified properly, FATS will produce accurate error locations and tape lengths.</p>   |
| <b>FILES=n</b>                   | <p>Applies to: READ</p> <p>"n" specifies how many physical files FATS is to read from its input tape. A physical file is defined as all records up to and including one tape mark. If the input tape contains labels, the labels before and after each data file will each count as one file, so each labeled file is actually three files. If "NOREWIND" is specified, the header label file of the first data file will not be read. If "n" is 0, FATS will terminate when it reads two consecutive tape marks (with no intervening data); this is usually sufficient to read all data files on the tape.</p> <p>Default: 3 files (one labeled file).</p> <p><b><i>WARNING: If "n" specifies more files than are present on the tape, or if "n" is 0 and the tape does not end in two tape marks, the tape may physically run off the end of the reel and require operator intervention to rethread it (3480 tape cartridges do not have this problem, but will get an error instead). Multi-volume data sets do not have double tape marks at the end except for the last volume. FATAR, the companion product to FATS, can properly read multi-volume tapes and is a better choice for most verification operations.</i></b></p> |
| <b>RETCODE</b>                   | <p>Applies to: ALL</p> <p>Causes FATS to terminate with a return code of 12 rather than abending in all cases where it would terminate with a U0888 abend code (due to major errors).</p> <p>Default: Abend with U0888 if major errors occur.</p>  |
| <b>REWIND</b><br><b>NOREWIND</b> | <p>Applies to: ALL</p> <p>"REWIND" (the default) causes FATS to rewind all tapes to load point before starting the specified operation, regardless of any "LABEL=" parameter in JCL. "NOREWIND" will tell FATS to leave the tape at the user specified position; for "WRITE" and "ERASE", this can be used to certify/erase the remainder of a tape beyond existing files. "NOREWIND" resets all label options.</p> <p>Default: REWIND</p>   |
| <b>WTO</b>                       | <p>Applies to: ALL</p> <p>Causes FATS to report the results of each operation on the system console via a one line message indicating the total number of permanent and temporary errors, or abnormal completion of the operation.</p>   |

CONTINUED . . .

## 14.3 CONTINUED . . .

**MODE=xx**

Applies to: ALL (except ANALYZE)

"xx" is a 2-digit hexadecimal number specifying a valid mode parameter for the tape drive being used. For 7-track drives, "MODE=xx" must be specified. For 9-track drives and IBM 3480 tape cartridge drives, it is optional and usually does not need to be specified. Valid values and their meanings are shown in the following table.

Default: For 9-track/3480, the highest density possible on the tape drive

| ===== VALID MODE VALUES ===== |               |            |            |                |                              |                   |               |  |
|-------------------------------|---------------|------------|------------|----------------|------------------------------|-------------------|---------------|--|
| VALID                         |               | DEFAULT    |            |                | -- 7-TRACK OPTIONS --        |                   |               |  |
| <u>MODE</u>                   | <u>TRACKS</u> | <u>BPI</u> | <u>GAP</u> | <u>BLKSIZE</u> | <u>TRANSLATE</u>             | <u>CONVERSION</u> | <u>PARITY</u> |  |
| 13                            | 7             | 200        | 150        | 4000           | OFF                          | ON                | ODD           |  |
| 23                            | 7             | 200        | 150        | 4000           | OFF                          | OFF               | EVEN          |  |
| 33                            | 7             | 200        | 150        | 4000           | OFF                          | OFF               | ODD           |  |
| 2B                            | 7             | 200        | 150        | 4000           | ON                           | OFF               | EVEN          |  |
| 3B                            | 7             | 200        | 150        | 4000           | ON                           | OFF               | ODD           |  |
| 53                            | 7             | 556        | 420        | 11120          | OFF                          | ON                | ODD           |  |
| 63                            | 7             | 556        | 420        | 11120          | OFF                          | OFF               | EVEN          |  |
| 73                            | 7             | 556        | 420        | 11120          | OFF                          | OFF               | ODD           |  |
| 6B                            | 7             | 556        | 420        | 11120          | ON                           | OFF               | EVEN          |  |
| 7B                            | 7             | 556        | 420        | 11120          | ON                           | OFF               | ODD           |  |
| 93                            | 7             | 800        | 600        | 16000          | OFF                          | ON                | ODD           |  |
| A3                            | 7             | 800        | 600        | 16000          | OFF                          | OFF               | EVEN          |  |
| B3                            | 7             | 800        | 600        | 16000          | OFF                          | OFF               | ODD           |  |
| AB                            | 7             | 800        | 600        | 16000          | ON                           | OFF               | EVEN          |  |
| BB                            | 7             | 800        | 600        | 16000          | ON                           | OFF               | ODD           |  |
| CB                            | 9             | 800        | 480        | 16000          |                              |                   |               |  |
| C3                            | 9             | 1600       | 960        | 32000          |                              |                   |               |  |
| D3                            | 9             | 6250       | 1875       | 32760          |                              |                   |               |  |
| D4                            | 18            | 38000      | 3040       | 32760          | -- 3480 TAPE CARTRIDGE DRIVE |                   |               |  |
| 03                            |               | 0          | 0          | 32760          | -- NON-SUPPORTED DEVICE      |                   |               |  |

## 14.3.7 ANALYZE PARAMETERS

All of the parameters which are valid on an "ANALYZE" statement input to FATAR (as described in a later section of this manual), are valid on a "DEFAULT" statement or an "ANALYZE(n)" statement when FATAR is invoked as a subtask of FATS.

## 16.0 FATS EXAMPLES OF USAGE

### INTRODUCTION

Examples are provided here to guide you in the use of FATS. As many of the common uses of FATS as possible have been included, and most of the control statements and parameters are illustrated here. However, be aware that they are just examples and must be customized to your installation and situation before use. Areas of such customization will probably include: data set names, unit names for tape, special parameters for tape management systems. The examples assume that no special "STEPLIB" DD statement is required to execute FATS; this may not be true in your installation.

Because of the many possible combinations of options, control statements, tape label types, etc., it is not practical to create examples which cover all potential FATS usage. Many examples illustrate more than one aspect of FATS use, e.g., two or more keywords together. This does not mean that the two must always go together unless the accompanying text says so.

Many examples use "LABEL=(,BLP)" on the tapes in order to be able to mount a labeled tape without knowing its volume serial or data set names. Sometimes this is simply a convenience, but for many FATS operations it is essential. In some installations, system parameters prevent the usage of "BLP" by most users; in this case, the FATS control statement parameter "BLP" can be substituted, although "BLP" may still be suppressed by installation exits at open time.

### 16.1 INDEX TO EXAMPLES

#### INDEX TO EXAMPLES

The first-time or infrequent user of FATS should review many of the examples below since they frequently build upon one another in illustrating the usage and effects of various parameters. For your convenience, however, here is a list of the examples provided to aid in quickly finding the one you need. It shows the title of the example and the operations and keywords it illustrates.

| #  | <u>EXAMPLE NAME</u>                                   | <u>OPERATION</u> | <u>KEYWORDS</u>                                    |
|----|---|------------------|--|
| 1  | Certify SL tapes                                      | WRITE            | MULT   |
| 2  | Certify SL tapes on multiple drives                   | WRITE            | VOL  |
| 3  | Certify and label tapes                               | WRITE            | VOL,VOLINCR,<br>MAXVOLN,MAXERR,<br>MAXCERR,NEWPAGE |
| 4  | Certify scratch tapes on 3 drives                     | DEFAULT<br>WRITE | MULT,MAXVOLN,MAXCLEAN                              |
| 5  | LABEL tapes   | DEFAULT<br>LABEL | LINECNT,OWNERID,VOL,<br>VOLINCR,MAXVOLN            |
| 6  | Label and certify new tapes                           | WRITE            | VOL,VOLINCR,MAXCLEAN,WTO                           |
| 7  | Visually examine bad tape                             | WRITE            | STOP,VOL   |
| 8  | Verify a SL tape                                      | ANALYZE          |  |
| 9  | Verify a multi-file SL tape                           | READ             | VOL,FILES  |
| 10 | Verify on several drives                              | DEFAULT<br>READ  | VOL,FILES  |
| 11 | Erase on multiple drives                              | DEFAULT<br>ERASE | VOLINCR,MAXVOLN,VOL                                |
| 12 | Re-initialize 3490E cartridges for use on 3480 drives | DEFAULT<br>LABEL | BLP,VOL  |

#### ISPF DIALOG SUPPORT

Most of the FATS functions can be performed by using the FATS ISPF interface. See [Section 93](#) for further details.

## 16.2 EXAMPLES

**EXAMPLE 1:** This example certifies any number of standard labeled tapes. Since no volume serial is specified, the operating system will ask for a scratch tape. With "LABEL=(,SL)" specified in JCL, tapes with any volume serials can be mounted. A tape management system, if present, can protect you by confirming the scratch status of any tape mounted. The "MULT" parameter will cause FATS to continuously request new scratch tapes until the system operator replies to the FATS console message or issues the operator 'P' command (See Section 13.1). Since the "SAVLAB" parameter is the FATS default, the original volume serial of each tape will be preserved. Since no "MODE=" parameter is specified, the tape will be certified at the highest density of which the tape drive is capable. The "PARM=I" on the EXEC statement causes FATS internal documentation to be printed on SYSPRINT.

```
//FATS          EXEC   PGM=FATS , PARM=I
//SYSPRINT     DD     SYSOUT=A
//SYSUDUMP     DD     SYSOUT=A
//TAPE1        DD     DSN=FATS , UNIT=TAPE ,
//              LABEL=( , SL ) , DISP=( , KEEP )
//SYSIN        DD     *
                DEFAULT   MODIFY=YES
                WRITE(1)   MULT
```

**EXAMPLE 2:** Certify 3 standard labeled tapes on two tape drives, specifying the required volume serials in FATS control statements via the "VOL=" parameter. "UNIT=(TAPE,,DEFER)" on the TAPE DD statements will prevent the operating system from requesting a mount until FATS passes it the required volume serial. Each tape must be previously labeled with the proper volume serial since FATS is opening them as labeled. Since the "SYSPRIN1" and "SYSPRIN3" DD statements are present, separate detail reports will be printed for each tape drive. Since the "SUMMPRT" DD statement is present, a separate summary report for all tapes on all drives will be printed.

```
//FATS          EXEC   PGM=FATS
//SYSPRINT     DD     SYSOUT=A
//SYSPRIN1     DD     SYSOUT=A
//SYSPRIN3     DD     SYSOUT=A
//SUMMPRT      DD     SYSOUT=A
//SYSUDUMP     DD     SYSOUT=A
//TAPE1        DD     DSN=FAT1 , UNIT=( TAPE , , DEFER ) , DISP=( , KEEP )
//TAPE3        DD     DSN=FAT2 , UNIT=( TAPE , , DEFER ) , DISP=( , KEEP )
//SYSIN        DD     *
                WRITE(1)   VOL=111111
                WRITE(3)   VOL=(222222 , 333333)
```

**EXAMPLE 3:** Twenty tapes, with volume serials "AX0001" to "AX0020" are to be labeled and certified. Their current internal volume labels are unknown so label processing must be bypassed ("LABEL=(,BLP)" in JCL). Parameter "VOL=AX0001" gives the first volume serial to be requested, parameter "VOLINCR=1" specifies that the serial is to be incremented by 1 each time, and parameter "MAXVOLN=20" limits FATS to 20 such volumes. The presence of the "VOL=" parameter (without other labeling parameters) causes the new tapes to be re-labeled. The parameter "NEWPAGE" causes the detail report for each tape to start on a new page (for easier separation), and the "MAXERR=5" and "MAXCERR=10" parameters will cause any given tape to terminate if it has more than 5 permanent errors or a 10 inch continuous error.

NOSAVE prevents reading blank tape and allows labeling to take place.

```
//FATS          EXEC   PGM=FATS
//SYSPRINT     DD     SYSOUT=A
//SYSPRIN5     DD     SYSOUT=A
//SUMMPRT      DD     SYSOUT=A
//SYSUDUMP     DD     SYSOUT=A
//TAPE5        DD     DSN=FATS , UNIT=( TAPE , , DEFER ) ,
//              LABEL=( , BLP ) , DISP=( , KEEP )
//SYSIN        DD     *
                WRITE(5)   VOL=AX0001 , VOL INCR=1 , NEWPAGE , NOSAVE ,
                MAXVOLN=20 , MAXERR=5 , MAXCERR=10
```

CONTINUED . . .

## 16.2 CONTINUED . . .

**EXAMPLE 4:  
CERTIFY  
SCRATCH  
TAPES ON 3  
DRIVES**

94 scratch tapes are to be certified on three tape drives. The system operator knows which tapes to mount, so FATS is to simply call for scratch tapes. The DEFAULT statement is used so that the other FATS parameters only need to be specified once. The "MULT" parameter will cause FATS to repeatedly call for new tapes on each drive (since no volume serial is given in the JCL, scratches will be requested). The "MAXVOLN=31" parameter causes each drive to terminate when 31 tapes have been certified on it (for the third drive, 32). "MAXCLEAN=10" instructs FATS to ask the operator (via a console message FATSW06) to clean each tape drive between every set of 10 tapes (i.e., 3 cleanings per drive in this example) so that dirty tape drives don't cause false errors. Lengths will be reported in metric.

**Note:** The "SAVLAB" parameter, which is the default, causes FATS to preserve the volume serials on the tapes; if any tape does not contain a label, FATS will ask the operator for the correct label.

```
//FATS          EXEC  PGM=FATS
//SYSPRINT     DD    SYSOUT=A
//SYSPRIN1     DD    SYSOUT=A
//SYSPRIN2     DD    SYSOUT=A
//SYSPRIN3     DD    SYSOUT=A
//SUMMPRT      DD    SYSOUT=A
//SYSUDUMP     DD    SYSOUT=A
//TAPE1        DD    DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//TAPE2        DD    DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//TAPE3        DD    DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//SYSIN        DD    *
                DEFAULT  MULT,MAXVOLN=31,MAXCLEAN=10,METRIC
                WRITE(1)
                WRITE(2)
                WRITE(3)  MAXVOLN=32
```

**EXAMPLE 5:  
LABEL TAPES**

Volumes "TST990", "TST994", and "TST998" are to be labeled with those volume serials. The "VOL=" parameter specifies the first of those serials. The "VOLINCR=4" parameter causes that serial (TST990) to be incremented. Although "MAXVOLN=50" is specified, the incrementing stops when the numeric portion of that serial cannot be incremented without overflowing. "LINECNT=80" requests 80 lines per page on all FATS reports.

```
//FATS          EXEC  PGM=FATS
//SYSPRINT     DD    SYSOUT=A
//SYSPRIN1     DD    SYSOUT=A
//SUMMPRT      DD    SYSOUT=A
//SYSUDUMP     DD    SYSOUT=A
//TAPE1        DD    DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//SYSIN        DD    *
                DEFAULT  LINECNT=80,OWNERID='TEST TAPE'
                LABEL(1)  VOL=(TST990),
                VOLINCR=4,MAXVOLN=50
```

CONTINUED . . .

## 16.2 CONTINUED . . .

**EXAMPLE 6:** The tape librarian has 250 brand new tapes to label and certify (certifying is recommended to clean loose oxide from the new tapes) starting at volume serial 500001. The "VOL=" and "VOLINCR=" parameters will cause this to happen. Since no "MAXVOLN=" parameter was specified, FATS will call for up to 682 volume serials (up to 500682). However, when all tapes are done, or if the certification process must be stopped to do other work, the librarian can stop FATS by replying to the FATS console message FATSW02. The "WTO" parameter will cause a 1-line summary of the errors on each tape to be written to the console so that potential problem tapes can be immediately identified. Since these are new tapes, the "MAXCLEAN=5" parameter requests tape drive cleaning after every five tapes to remove accumulated oxide.

NOSAVE prevents reading blank tape and allows labeling to take place.

```
//FATS          EXEC   PGM=FATS
//SYSPRINT     DD     SYSOUT=A
//SYSPRIN1     DD     SYSOUT=A
//SUMMPRT      DD     SYSOUT=A
//SYSUDUMP     DD     SYSOUT=A
//TAPE1        DD     DSN=FATS,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),DISP=(,KEEP)
//SYSIN        DD     *
                WRITE(1)  VOL=500001,VOLINCR=1,MAXCLEAN=5,WTO,NOSAVE
```

**EXAMPLE 7:** The tape librarian wants to visually examine the bad spots on a known bad tape. The "STOP" parameter will cause FATS to stop and issue a message to the console when permanent data checks are encountered, at which time the tape can be examined. The operator may reply to the message, telling FATS to terminate, continue (and stop again on the next error), or turn off "STOP MODE" and continue normally. Stop mode might be used to determine if tape errors are due to damage or simply due to removable contamination.

```
//FATS          EXEC   PGM=FATS
//SYSPRINT     DD     SYSOUT=A
//SYSPRIN1     DD     SYSOUT=A
//SUMMPRT      DD     SYSOUT=A
//SYSUDUMP     DD     SYSOUT=A
//TAPE1        DD     DSN=FATS,UNIT=(TAPE,,DEFER),DISP=(,KEEP)
//SYSIN        DD     *
                WRITE(1)  VOL=111111,STOP
```

**EXAMPLE 8:** A standard labeled tape "22222" is to be verified for readability. The FATS companion product "FATAR" is to be used to do so. Since FATAR is opening it as labeled, the proper data set name and volume serial must be given. Since the "FILES=" parameter is not given, FATAR will read to the end of the tape (2 tape marks or EOVL labels).

```
//FATS          EXEC   PGM=FATS
//SYSPRINT     DD     SYSOUT=A
//SYSPRIN7     DD     SYSOUT=A
//SUMMPRT      DD     SYSOUT=A
//SYSUDUMP     DD     SYSOUT=A
//TAPE7        DD     DSN=DATA.SET.NAME,UNIT=TAPE,
//              VOL=SER=22222,DISP=OLD
//SYSIN        DD     *
                ANALYZE(7)
```

CONTINUED . . .

## 16.2 CONTINUED . . .

**EXAMPLE 9:** A standard labeled tape "333333" containing two files is to be certified for readability. The volume serial is specified via the FATS parameter "VOL=", but a dummy serial must be given on the TAPE DD statement to satisfy operating system requirements; "UNIT=(TAPE,,DEFER)" keeps the system from requesting the mount until FATS sets the actual serial. "LABEL=(,BLP)" is specified since the true data set name of the first file is unknown. "FILES=6" causes two standard label files to be read (the header and trailer label file count as files to FATS).

```
//FATS          EXEC  PGM=FATS
//SYSPRINT      DD    SYSOUT=A
//SYSPRIN1      DD    SYSOUT=A
//SUMMPRT       DD    SYSOUT=A
//SYSUDUMP      DD    SYSOUT=A
//TAPE1         DD    DSN=DATA.SET.NAME,UNIT=(TAPE,,DEFER),
//              VOL=SER=DDDDDD,DISP=OLD,LABEL=(,BLP)
//SYSIN         DD    *
READ(1)        VOL=333333,FILES=6
```

**EXAMPLE 10:** Several tapes are to be verified for readability using two tape drives. The "VOL=" parameter specifies which tapes are to be mounted on each drive. "FILES=0" tells FATS that each tape is to read until two consecutive tape marks are encountered (two tape marks normally indicates end of data on the tape).

**WARNING:** Tapes which are part of a multi-volume set (except the last) DO NOT end in two tape marks and may run off the end of the reel if read with "FILES=0". FATAR is recommended to verify such tapes.

```
//FATS          EXEC  PGM=FATS
//SYSPRINT      DD    SYSOUT=A
//SYSPRIN1      DD    SYSOUT=A
//SYSPRIN2      DD    SYSOUT=A
//SUMMPRT       DD    SYSOUT=A
//SYSUDUMP      DD    SYSOUT=A
//TAPE1         DD    DSN=ANY.NAME,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),VOL=SER=DUMMY1,DISP=OLD
//TAPE2         DD    DSN=ANY.NAME,UNIT=(TAPE,,DEFER),
//              LABEL=(,BLP),VOL=SER=DUMMY2,DISP=OLD
//SYSIN         DD    *
              DEFAULT  FILES=0
              READ(1)  VOL=(111111,222222,333333)
              READ(2)  VOL=(444444,555555,666666)
```

CONTINUED . . .

## 16.2 CONTINUED . . .

**EXAMPLE 11:** A large number of tapes are to be "Data Security Erased" before they are sold. Since they have consecutive serial numbers, the FATS parameters "VOL=", "VOLINCR=1", and "MAXVOLN=50" are used to erase 50 tapes on each of 3 tape drives. FATS will call for the tapes by serial as needed. Although "LABEL=(,BLP)" has been specified to suppress operating system label handling, FATS will re-label each tape as it erases it. Note that "Data Security Erase" operations do not tie up the tape control unit, so that each of the three erase operations can proceed at full tape speed, and other tape jobs will not be impacted.

```
//FATS      EXEC  PGM=FATS
//SYSPRINT  DD    SYSOUT=A
//SYSPRIN1  DD    SYSOUT=A
//SYSPRIN3  DD    SYSOUT=A
//SYSPRIN5  DD    SYSOUT=A
//SUMMPRT   DD    SYSOUT=A
//SYSUDUMP  DD    SYSOUT=A
//TAPE1     DD    DSN=FATS,UNIT=(TAPE,,DEFER),
//          LABEL=(,BLP),DISP=(,KEEP)
//TAPE3     DD    DSN=FATS,UNIT=(TAPE,,DEFER),
//          LABEL=(,BLP),DISP=(,KEEP)
//TAPE5     DD    DSN=FATS,UNIT=(TAPE,,DEFER),
//          LABEL=(,BLP),DISP=(,KEEP)
//SYSIN     DD    *
            DEFAULT VOLINCR=1,MAXVOLN=50
            ERASE(1) VOL=100001
            ERASE(3) VOL=100051
            ERASE(5) VOL=100101
```

**EXAMPLE 12:** You need a fair number of blank tapes to copy backups and data to take to a disaster recovery site which does not have 3490E tape sub-systems. You notice that the majority of your scratch cartridges are in 3490E format and need to be re-initialized to avoid excessive intervention. To expedite matters you set aside a batch of tapes and obtain their serial numbers This jobstream will allow you to re-initialize the scratch tapes set aside for this purpose. Since your JES setup forces BLP coded in JCL to become NL, the BLP parameter is supplied as a DEFAULT parameter. In combination with DEFER, the LABEL option is changed by FATS to BLP. If this is not done, the operator will need to supply the serial number for each reel at the console. If you have a security system that prohibits use of BLP, you may need to obtain the authority to use it for this purpose as FATS BLP option will not circumvent site security. The coding of the list of volume serial numbers illustrates the use of FATS control card continuation. A comma followed by a blank indicates that the control statement is continued on the next card.

```
//FATS      EXEC  PGM=FATS
//SYSPRINT  DD    SYSOUT=A
//SYSPRIN1  DD    SYSOUT=A
//SUMMPRT   DD    SYSOUT=A
//SYSUDUMP  DD    SYSOUT=A
//TAPE1     DD    DSN=FATS,UNIT=(3490,,DEFER),
//          DISP=(,KEEP)
//SYSIN     DD    *
            DEFAULT BLP
            LABEL(1) VOL=(701034,700987,700986,700899,700843,
                          700824,700805,700713,700592,700321,
                          700317,700115)
```

# FATS<sup>®</sup> & FATAR<sup>™</sup>

USER DOCUMENTATION

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## 20.0 FATAR FUNCTIONAL DESCRIPTION

**WHAT IS FATAR?** FATAR (Fast Analysis of Tape And Recovery) is a multi-purpose magnetic tape utility, which can be executed under MVS, MVS/XA, and MVS/ESA. FATAR can read any magnetic tape which is usable by those operating systems, and can process multiple files and multiple volumes in one execution. It offers the ability to summarize or examine in detail the data read from the tape. FATAR's error recovery procedures are often capable of reading data which causes permanent data checks when read by normal access methods. FATAR can also create a copy of the input data on another tape, optionally applying user-specified modifications as it copies.

In most installations, much of the installation's data is stored on magnetic tape, which is subject to many hazards. This data will be useful only if it was written correctly when first recorded and is still readable when needed. Many hours of man and computer time can be wasted trying to recover from physical (hardware) or logical (data) errors on tape. FATAR can greatly reduce this recovery time, or even discover problems before they occur.

In addition, FATAR's utility functions for reading and copying tape have many uses.

**WHO USES FATAR?** FATAR is ideal for use by tape librarians and operations personnel, by applications programmers, and by systems programmers.

**WHAT IS FATAR USED FOR?** FATAR has many applications, including:

- Investigating an "unknown" tape, to discover its label type, file count, DCB characteristics, etc.
- "MAPPING" a tape, providing a compact summary of the characteristics of all files on a tape.
- Examining the data on a tape.
- Verifying that a tape file is properly formatted (every block is checked against its DCB information).
- Detecting and correcting invalid variable spanned records.
- Verifying that certain data fields contain valid data.
- Scanning a tape for a certain type of data.
- Scanning a tape for temporary and permanent read data checks, which may cause application job failures.
- Verifying the readability of critical or archive tapes.
- Correcting invalid data or data checks by creating a a copy of the input tape(s) with the bad data corrected or dropped.
- Creating a good copy of a tape which was not properly closed (such as during a system failure).
- Recovering data from a tape partially overwritten in error with a shorter file.
- Creating a backup copy of any tape (or multi-volume set of tapes), even if multiple files exist on the tape.
- Replacing tapes with small blocksizes with copies using a larger blocksize (reducing elapsed and CPU times).
- Creating a copy of a tape with a different label type from the original (such as AL-to-SL or NL-to-SL).
- Creating copies of tapes at higher density or on a different type of tape drive (such as the IBM 3480).
- Copying FDR/ABR tapes (similar to the operation of the FDR utility FDRTCOPY) or other tapes with blocksizes over 32K.

CONTINUED . . .

## 20.0 CONTINUED . . .

**FATAR CONTROLS** FATAR has a flexible set of command statements entered as 80-character records, of which only columns 1-71 are used by FATAR. For many operations, the commands required are very simple since the default operations of FATAR are designed to handle the most common user requirements. However, the FATAR statements allow you to greatly modify FATAR operation, including the ability to specify an operation down to a specific block and record on the tape if required.

**INPUT TAPE PROCESSING** Unless you instruct it otherwise, FATAR will read all files on all tape volumes specified on its TAPEIN DD statement. If DCB information is available (from JCL or from the tape labels), FATAR will deblock each tape block into individual records if the record format is fixed or variable, allowing you to examine or modify each logical record. If you prefer, or if DCB information is not available, FATAR will treat each tape block as one record.

Since most tapes in IBM installations are labeled, FATAR will normally expect input tapes to contain labels (either IBM standard or ISO/ANSI), and will intelligently print the labels, extract data set name and DCB information from them and treat only real data files as data. This will be true even if you have specified "LABEL=(,BLP)" in JCL, so that you can mount tapes whose volume serial you don't know or don't care about and still print and process their labels. However, FATAR will recognize unlabeled tapes automatically and treat all files on them as data. If you have a need to process label files as data, you can instruct FATAR to do so.

If the input data is in ASCII, FATAR will translate it to EBCDIC before processing.

If hardware data checks occur reading a block from TAPEIN, FATAR will perform a variety of RETRY operations to attempt to successfully perform the read. Even if all recovery is unsuccessful, FATAR will still make the block as read available for manual correction.

**DATA BLOCK PROCESSING** Once a block has been read, FATAR offers facilities via FATAR commands to:

- Print the block (in EBCDIC, HEXADECIMAL, or DUMP format (HEX and EBCDIC together) ). All or part of the block may be printed.
- Change the length of the block (extra bytes may be added at the beginning or end of the block or both).
- Change data in the block (either at a specified location or by scanning for a string).
- Scan specified fields in the block for valid packed or zoned decimal numbers (to help eliminate data exceptions in application programs).
- Scan specified fields in the block for validity by comparison to a user specified table of valid or invalid values.
- Scan for specific strings of data, in specified locations or in any location.
- Drop data (prevent copy to output tape).

If logical records are being deblocked, all of the above (except block length changing) can be applied to each logical record.

**NOTE:** Changes to the input blocks never affect the input tape. Modified blocks are written only to the output tape, if present.

CONTINUED . . .

**20.0 CONTINUED . . .**

**OUTPUT TAPE PROCESSING** If the output tape DD statement TAPEOUT is present, FATAR will copy each file read from TAPEIN to the output tape. This will be an exact copy of the input, unless you request modification of one or more data blocks, or dropping of certain data blocks. You may also request that entire files be dropped from the copy.

By default, any block which had an unrecoverable data check when read from TAPEIN will not be written to TAPEOUT, but you may override this.

FATAR will issue a standard OPEN and CLOSE for each file on TAPEOUT, so that security systems and tape management systems will be properly invoked as long as they interface with OPEN/CLOSE.

If the output tape is in ASCII format, the data blocks will be converted from EBCDIC to ASCII before they are written. If the records are variable format, appropriate conversion to or from the ASCII variable format and the IBM format will be done.

## 21.0 FATAR TECHNICAL SUMMARY

### 21.1 GENERAL

**THE FATAR PROGRAM** FATAR (Fast Analysis of Tape And Recovery) is a utility program for reading and copying magnetic tapes.

FATAR requires 256K of memory. If reblocking is specified (the REBLOCK= keyword on the ANALYZE statement), the size of the reblocking buffer (the value of the REBLOCK= keyword rounded up to a multiple of 4K) must be added to the FATAR memory requirements.

Although FATAR is normally LINKEDITed as an authorized program under MVS, it requires authorization only if the "BLP" control statement keyword is used, if 3480 tape cartridge drives are used as input, or if the ability to automatically answer CA-1 (UCC1, TMS) tape control messages is desired.

FATAR may also be called as a sub-task of its companion product FATS. This mode of operation allows multiple FATAR sub-tasks to operate in the same job, and also allows use of FATS keywords for specifying multiple executions of FATAR using the same input tape drive, but with different input tapes. In this case the memory requirements of FATAR will be the values listed above times the number of concurrent FATAR sub-tasks (plus the memory requirements of FATS itself, detailed elsewhere in this manual).

**CONTROL AND REPORT I/O** FATAR accepts its control statement input from DD statement SYSIN. If no SYSIN DD is present in the step, all defaults are assumed. The control statements are 80-byte records, of which only columns 1 to 71 may contain information.

Report output is directed to DD statement SYSPRINT, which may be a SYSOUT file or any tape or disk file valid for output. TAPESUMM is an optional additional printout file, whose use is described below. Both files always have logical record lengths of 121 bytes. If they are directed to SYSOUT or if the blocksize is unspecified, 121 is assumed. The user may specify a blocksize for tape or disk output, but if it is not a valid multiple of 121, 1210 will be assumed.

If FATAR is run as a sub-task of FATS, these differences apply: printout will be directed to SYSPRINx (where "x" is the same as the digit on the FATS control statement "ANALYZE(x)") and "TAPESUMM" will be ignored if present. Most options are passed from FATS, but the user can override the options or provide MODIFICATION/SCAN statements by supplying them in a DD statement SYSINx ("x" same as above).

### 21.2 INPUT TAPE PROCESSING

**FATAR INPUT TAPE I/O** FATAR uses EXCP rather than any standard access method when reading input tape (from DD name TAPEIN) in order to have more control over error recovery processes. FATAR will issue a normal OPEN for the first file on each input volume, allowing security and tape management systems to perform their normal checking and authorization. However, if a tape volume contains more than one file, FATAR will switch files without any intervening CLOSE/OPEN. If a data set occupies more than one input tape volume, FATAR will CLOSE at the end of the current volume and re-OPEN for the next volume. This allows FATAR to process any multi-file and/or multi-volume tape or tapes in one execution.

FATAR supports all tape drives which are supported on its host operating systems, including IBM 2400 series drives, 3400 series drives, 3480/3490 tape cartridge drives, and all plug compatible tape hardware. Both 7-Track and 9-Track drives are supported, and all densities are supported as well, from 200 BPI 7-Track to 6250 BPI 9-Track. The IBM 3480 tape sub-system is supported (at 38000 bpi) in 3420 compatibility, full function, and IDRC modes.

If FATAR is invoked as a sub-task of FATS, it will read its input from DD name TAPEx rather than TAPEIN, where "x" is the same as the digit on the FATS control statement "ANALYZE(x)". This is compatible with the FATS DD name.

CONTINUED . . .

## 21.0 CONTINUED . . .

**EXTENDED  
I/O ERROR  
RECOVERY**

I/O errors on a tape are often fatal to an application program. If the operating system cannot recover from an error, the program has little choice except to abort and leave it up to a human to investigate and correct the problem. FATAR is an ideal tool for fixing these problems due to its extended error recovery and its response to unrecoverable errors.

FATAR's extended I/O error recovery attempts to RETRY most errors in much the same way that the operating system's error recovery does. However, FATAR has the capability of retrying a read operation up to 256 times in both the forward and backward directions. Operating system error recovery does less than 1/6 of this and will not always RETRY in the backwards direction (particularly for large blocks).

To further improve recovery, FATAR will execute a special tape cleaner action on every fourth retry, moving the tape rapidly back and forth across the cleaner blade built-in to the tape drive, in order to attempt to dislodge any dirt or loose oxide on the tape surface.

Sometimes, the nature of an error on a particular data block is such that any attempt at recovery causes the tape to be mis-positioned, resulting in either lost data or an error recovery loop. Operating system error recovery cannot be totally shut off, and therefore cannot recover from this situation. FATAR can be instructed to do absolutely no recovery actions during a given run, and can often successfully bypass the bad block and recover the remainder of the data on the tape.

If FATAR's error recovery actions are unsuccessful, the data block is considered to be a permanent error. By default, such blocks will be dropped (not written) if you are copying to an output tape. However, FATAR gives the option of examining what data it was able to read and of modifying or replacing the invalid data. Most of the FATAR command statements can be specified to only apply to blocks which had a permanent data check.

Since FATAR must inhibit normal operating system error recovery in order to control its own error recovery actions, any errors encountered by FATAR will not be recorded in the operating system "SYS1.LOGREC" (EREP File) or in SMF Type 21 (ESV – Error Statistics by Volume) records. For cartridge drives, the total errors encountered by the tape drive will still be recorded on LOGREC and SMF, but individual errors will not.

**INPUT TAPE  
LABEL  
PROCESSING**

By default, FATAR will check for the presence or absence of labels on the input tape, although it can be instructed not to look for labels. This label processing will happen regardless of the label type specified in JCL, so that it is possible to specify "LABEL=(,BLP)" on the DD statement, mount any labeled tape (even if its volume serial is unknown), and have FATAR still read and act on its labels.

FATAR supports both standard IBM labels ("SL" labels), and ISO/ANSI labels ("AL" labels – both version 1 and 3). FATAR will recognize which type of labels the tape has and perform the proper translation and conversion. If ANSI labels are found, it is assumed that the associated data file is written in ASCII code, and an ASCII-to-EBCDIC translation will be done.

When reading header labels, FATAR will extract the data set name (only the last 17 characters of the name are recorded in the label), and DCB information (RECFM, LRECL, and BUFOF(ANSI only)). These values will be used for opening the output tape file and for deblocking the blocks of the following data file.

When reading trailer labels, FATAR will verify the block count against the actual count of blocks read in the preceding data file, and produce a warning message if they do not match.

CONTINUED . . .

## 21.0 CONTINUED . . .

**INPUT TAPE  
DATA BLOCK  
PROCESSING**

As FATAR begins processing each data file, it analyzes the DCB information (record format and logical record length) it has available. As described above, if labels are being processed, FATAR will extract this DCB information from them. However, if DCB parameters were supplied on the DD statement for the input tape, they will override the tape labels. This allows you to suppress FATAR deblocking operations (by specifying "DCB=RECFM=U") or to correct invalid DCB information on the tape labels.

If the record format of the file is fixed (RECFM=F or FB) or variable (RECFM=V, VB, or VBS), FATAR will attempt to deblock each data block it reads from the input tape into its logical records. Note that FATAR does not depend on the blocksize from the labels or JCL but will read all the data in each tape block and then attempt to deblock it according to the format and logical record length found. In fact, if the records are variable length, the record length from the labels/JCL is also not used, but rather the length fields at the start of each record are used to locate the records. If the record format is variable spanned, FATAR will deblock each record segment individually.

If FATAR finds an error during deblocking (which may be a fixed length block which does not contain an exact number of full records, or a variable length block whose last record is too long for the length remaining in the block), FATAR will print an error message and treat that block as unblocked for printing. This may aid in identifying improperly formatted tapes, but remember that blocks that were read with data checks may cause these errors simply because the data is invalid.

When deblocking is in effect, FATAR will not print or make available for modification the block prefix (for ASCII Tapes) or the block descriptor word ("BDW" – for variable format blocks). However, each record (including the record descriptor word or "RDW" for variable format) is printed and available for modification. The FATAR control statements "SCAN" and "REPLACE" (which validate and modify data respectively), will operate on individual records within the block, and can refer to specific records in specific blocks. Other FATAR statements ("PRINT", "DROP", and "KEEP") always operate on whole blocks.

If deblocking is suppressed or not possible, then the entire contents of each block (including any prefix or BDW) is printed or available for modification as if it were one single record.

### 21.3 MODIFICATION AND SCANNING

#### MODIFICATION AND SCAN STATEMENTS

FATAR MODIFICATION/SCAN Control Statements are edited by FATAR and stored in a table at initialization. As each block is read by FATAR, this table is scanned to see if any of these statements apply to the current block.

MODIFICATION/SCAN statements allow you to:

- Print the contents of any block
- Drop a block or record (suppress it from being written to the output tape)
- Keep a block (override FATAR's automatic drop of blocks which get data checks)
- Change the length of a block, making it shorter or longer than the original, and reposition the original data in the new block
- Replace data at a given position in a block or record, optionally verifying the original contents first
- Scan a block or record for any field containing a given value, and replace that with a new value (for either the first or every occurrence in the block or record)
- Validate that certain fields in a block or record contain valid packed or zoned (printable) decimal data
- Validate that certain fields in a block or record contain valid data according to a user-specified test table (which can specify either all of the valid byte values or all invalid byte values).

More than one modification may apply to a given block or record, so FATAR will scan the table for all applicable control statements, executing them in the order that they originally appeared in the control statement input. Also, one statement may apply to more than one block or record; a range of blocks, or a range of records within a block may be specified, or a statement may apply to all blocks and records in a file (to provide a global MODIFICATION/SCAN facility).

Normally, all blocks which are modified will be printed after modification, although this can be suppressed. If fields are being validated (the "SCAN" statement), FATAR will PRINT a warning about each field which fails validation. In any case, a message will be printed about every block which was affected by a control statement (documenting that it was "DROPPED", "PRINTED", etc.). If the control input contains MODIFICATION/SCAN statements which are never selected by FATAR (if, for example, they specify a block number which does not appear on the tape), they will be identified at the end of the FATAR run so that the user can investigate the reason.

Although the "DROP", "KEEP" and "REPLACE" statements are most meaningful when an output tape is being produced, they will be processed normally even when no output tape is present. This allows you to "Dry run" your control statements before you actually create an output.

## 21.4 OUTPUT TAPE PROCESSING

### OUTPUT TAPE PROCESSING

If the DD statement TAPEOUT is present, a logical duplicate of the input tape(s) will be generated. It is a "Logical" duplicate since FATAR will allow the copy to have a different type of labels from the original, tape density may be different, and user-specified modifications may be applied. The copy will normally have the same number of files as were found on the input tapes, in the same order, with the same data set names. It is possible to have FATAR create an "IMAGE COPY" of an input tape volume, an exact copy including all labels, in order to replace a known bad tape with a good one.

As each file is encountered on the input tape, the equivalent file will be opened on the output tape; however, it is possible to request FATAR to drop an entire file ("DROP LF=1,B=ALL") in which case the output file will not be opened at all. If labels are being processed on the input, the data set name and DCB information read from the header labels of the file will be used to open the output file (only the last 17 characters of the name are recorded in the header label and are available to FATAR), otherwise the data set name on the TAPEOUT DD Statement is used (DCB information will be copied from the TAPEIN DD statement, if present).

The user open exit may also be used to modify the data set name for the output file. If specified by EXIT=(OPENEXIT,exitname), the module "exitname" will be loaded and will be called before opening each output file copied to TAPEOUT. If "exitname" cannot be found FATAR will abend S806 upon parsing the EXIT= keyword statement.

The register conventions when the exit is called are:

R1 – Pointer to PLIST containing:

+0 Pointer to fullword code: 4 is normal call

8 is last call: clean up.

+4 Pointer to double word on double word boundary that can be used by the exit routine for any purpose such as remembering the subpool, length, and address of any area the exit acquired for operation.

+8 Pointer to TAPEIN JFCB

+12 Pointer to TAPEOUT JFCB

R13 – Pointer to a standard save area

R14 – Return address

R15 – Entry point of exit routine

The registers should be saved and restored by the exit routine. Only the TAPEOUT JFCB may be safely modified.

A sample exit routine (named "OPENEXIT") is included in file 3 of the installation tape. The sample exit replaces the high level qualifier (or everything in front of the first period) of the tape data set name with a new qualifier hard coded in the exit. This shell could be expanded to query the system catalog and/or the tape management catalog and to cause the output to be cataloged in either or both of them.

After all modifications have been applied to a given block from the input tape, it will be written to the output tape (unless it is being dropped). Standard operating system error recovery is permitted on the output tape, so any WRITE error reported back to FATAR will cause the entire run to be aborted at that point. If the output tape is in ASCII characters ("LABEL=(,AL)" or "DCB=OPTCD=Q" in JCL), the block will be translated to ASCII and its format converted (if variable).

Data set reblocking may be requested by the REBLOCK= keyword on the ANALYZE statement. If so, FATAR will reblock each copied file if it is fixed or variable format (except variable spanned and ANSI variable), if its logical record length is less than the new blocksize, and if its current blocksize is not greater than the value of MAXRBLK= (maximum blocksize to reblock) if specified. When reblocking, each logical record is copied into a reblocking buffer which is written to TAPEOUT only when full (or at end-of-file).

If FATAR is invoked as a sub-task of FATS, it will WRITE its output to DD name TAPE<sub>x</sub>OUT (if present) rather than TAPEOUT, where "x" is the same as the digit on the FATS control statement "ANALYZE(x)".

### TAPE SUMMARY

If DD statement TAPESUMM is present, FATAR will print a summary of each tape, detailing the data set name, DCB and label information, blockcount, bytecount, minimum, maximum, and average blocksizes, and approximate length in feet for each file read from the input tape.

## 22.0 FATAR OPERATIONS

### 22.1 INPUT TAPES AND LABELS

**INPUT TAPE** The input to FATAR may be any one of the following:

**TYPES**

- A standard labeled tape containing one or more files (data sets).
- A multi-volume set of standard labeled tapes containing one or more files.
- A tape with ISO/ANSI labels containing one or more files in ASCII.
- A multi-volume set of ISO/ANSI labeled tapes containing one or more files in ASCII.
- A cartridge in IDRC (compressed) format.
- a multi-volume set of IDRC (compressed) cartridges containing one or more files.
- An unlabeled tape containing one or more files, recorded in EBCDIC or ASCII.
- An unlabeled multi-volume set containing only one file (unlabeled multi-volume multi-file tape sets cannot be supported by FATAR since it cannot tell how many files are on each tape).

Labeled tapes are handled by FATAR independently of the "LABEL=" parameter in the JCL. In other words, the input TAPE DD statement may say "LABEL=(,SL)", "LABEL=(,AL)", or "LABEL=(,BLP)" and FATAR will still be able to recognize and process the labels on the tape. Standard labels (SL) and ANSI labels (AL) will be detected even when operating system label processing is bypassed. "LABEL=(,BLP)", if permissible in your installation, is often an advantage since it allows you to mount a tape whose volume serial or first data set name are unknown, or to set up a standard FATAR procedure which does not need to be modified for every run.

Conversely, unlabeled tapes or labeled tapes may be processed as unlabeled by FATAR so that all files on the tape (including files containing labels) are handled as data by FATAR, even if "LABEL=(,SL)" or "LABEL=(,AL)" is specified in the JCL (in this case, the tape is positioned so that the first file processed is the header labels of the file specified in the JCL). If the tape is truly unlabeled, FATAR will recognize that the first block read is not a label and treat the tape as unlabeled automatically.

If the "LABEL=" parameter on the TAPEIN DD statement specifies a file other than the first (e.g., "LABEL=5" or "LABEL=(4,BLP)"), FATAR will start its processing with that file. If the input tape is a labeled tape but you have specified "BLP" in the JCL, the file number must point FATAR to the header labels of some file on the tape (header labels normally start at files 1, 4, 7, etc.).

CONTINUED . . .

## 22.0 CONTINUED . . .

**INPUT FILE HANDLING** For labeled tapes, FATAR will process all of the files on all of the volumes specified on the input TAPE DD statement unless you deliberately limit it to a certain number of files with the "NUMFILES=" or "ENDAFTER=" parameters. FATAR will continue to read and process files until either:

- Two consecutive tape marks are read, with no data or labels between. On all labeled tapes, this indicates the end of the tape or multi-volume tape set.
- The number of files specified by "NUMFILES=" has been read.
- The block specified by the "ENDAFTER=" parameter has been read.
- "EOV" labels at the end of a tape are read (indicating that the data set is continued on another tape), but no more tape volume serials were specified.

This means that, when in label mode, FATAR can read and copy an entire multi-file, multi-volume tape set automatically.

For unlabeled tapes, FATAR will process multiple files on the input tape, but is unable to switch to new volumes since there is no way to tell how many files are on the current volume, and therefore when to switch volumes. By default, FATAR will process all files on one tape volume until two consecutive tape marks are read (as for labeled tapes). FATAR will also stop when it encounters "EOV" labels (even though labels are not being processed) since EOV labels are followed by only one tape mark; this prevents FATAR from running off the end of the tape. For a truly unlabeled tape which is part of a multi-volume set, you must tell FATAR how many files exist on the tape (via the "NUMFILES=" parameter) or it may run off the end of the reel.

**OS LABEL MODE** FATAR has a special label mode, "LABELS=OS", where all label processing and tape positioning is left to the operating system. In this mode, FATAR will handle exactly one file ("NUMFILES=1" is forced), but that file can be multi-volume, either labeled or unlabeled, and can even be several files concatenated together, as long as all the files are tape. "LABELS=OS" can be used to bypass the restriction on multi-volume unlabeled tapes, or to create one output tape file from multiple input files.

## 22.2 MODIFICATION AND SCANNING

**MODIFICATION AND SCAN PROCESSING** If FATAR MODIFICATION/SCAN statements were specified for a FATAR execution, FATAR will examine its table of such statements after each block is read for any such statements which apply to the current block. If more than one does apply, they will be executed in the order in which they appeared in the control statement input. If blocks are modified they will be printed (up to the length specified by the "PRTLEN" parameter) unless you suppress this printing with the "MSGLEVEL=0" parameter.

## 22.3 OUTPUT TAPES

**OUTPUT FILE PROCESSING** If the TAPEOUT DD statement is present, FATAR will copy each file read from the input to the output tape. During the copy operation, records, blocks, or even entire files may be omitted from the output tape by use of the "DROP" control Statement.

If labels are being processed on the input (LABELS=YES is specified or defaulted and the input is labeled), each output file will be opened with the data set name of the original file. Note that only the last 17 characters of the original name are available to FATAR unless the input DD Statement has the full name after DSN= and it matches the tape label in the last 17 positions.

If labels are not being processed on the input (LABELS=NO or LABELS=OS is specified or the input is not labeled), the data set name given on the TAPEOUT DD Statement will be used.

Similarly, if processing labels on TAPEIN, the DCB (record format, logical record length, and blocksize) from the labels of each file will be copied to the output tape. If input labels are not being processed then any DCB information given on the TAPEIN DD Statement will be used. The DCB blocksize may be modified for the output if reblocking (REBLOCK=) is chosen. You can specify a density on the TAPEOUT DD statement (via "DCB=DEN=n") in order to write a tape at other than the drive's default density; the density of TAPEIN and TAPEOUT do not need to be the same (TAPEOUT will default to the highest density possible on the tape drive assigned).

Regardless of the label type of the input tape, labels on the output tape are controlled by the "LABEL=" parameter on the TAPEOUT DD statement. You may say "LABEL=(,SL)", "LABEL=(,AL)", "LABEL=(,NL)", or "LABEL=(,BLP)" and the indicated label type will be generated on the output. This allows you to use FATAR to easily convert unlabeled tapes to labeled, labeled to unlabeled, ANSI labels to IBM, etc.

If labels are present on both TAPEIN and TAPEOUT, the expiration date will be copied from TAPEIN to TAPEOUT for each file processed. The expiration date can be overridden by placing the "EXPDT=" or "RETPD=" JCL parameter on the TAPEOUT DD statement. The creation date on TAPEOUT will be the current date unless the "CRDATE=COPY" parameter is used (not recommended for CA1 (UCC1, TMS) tape management system users).

If the "LABEL=" parameter on the TAPEOUT DD statement specifies a file other than the first (e.g., "LABEL=3" or "LABEL=(5,NL)"), FATAR will copy the first input file starting at that location on the output tape.

TAPEOUT may be a different device type from TAPEIN. This may be used to convert 7-Track to 9-Track, 1/2 inch tape to 3480 cartridge, etc.

**23.0 FATAR EXECUTION JCL**

To execute FATAR, the following JCL statements are required:

**ISPF PANEL SUPPORT** Extensive ISPF panel support is available for FATAR.  
See [Section 93](#) for further details on "How to Use the FATAR ISPF Panels".

**JOB STATEMENT** User supplied.

**STEPLIB/ JOBLIB DD STATEMENT** A STEPLIB or JOBLIB DD statement will be required if FATAR has been linked into a private library. It can be omitted if FATAR is in a system library which can be accessed without a STEPLIB/ JOBLIB statement (that is, a library in the system link list such as "SYS1.LINKLIB").

**EXEC STATEMENT** The EXEC statement specifies the FATAR program name, memory requirements (if your installation defaults are insufficient), and parameter field.

The memory requirements of FATAR are 128K, but this value may need to be adjusted up or down if the "SIZE=", "BLKSIZE=" or "REBLOCK=" keywords are used.

The parameter field (PARM=) is used to define to FATAR the amount of memory required for storing FATAR modification and scan control statement information. FATAR requires about 30 bytes per control statement plus the data length of any "D=", "V=", or "S=" data fields given. The default table size of 4096 bytes is sufficient for most FATAR applications, but if it proves to be inadequate, override it by "PARM='SIZE=nnnnn'"; where nnnnn can be 1-999999.

Examples :

```
//FATAR EXEC PGM=FATAR,REGION=128K
//FATAR EXEC PGM=FATAR,PARM='SIZE=8192',
// REGION=132K
```

**SYSPRINT DD STATEMENT** SYSPRINT receives all FATAR printout and reports (except the tape summary), and is normally allocated to a SYSOUT data set. Its DCB attributes are RECFM=FBA,LRECL=121. If blocksize is specified it must be a multiple of 121, otherwise it will default to 121 for SYSOUT or 1210 for other devices.

**TAPESUMM DD STATEMENT** TAPESUMM is optional, but if present receives a summary of all files on each tape processed, including data set name, tape label information, block and byte counts, footage length of each file, minimum, maximum and average block sizes, and read error counts. Its DCB attributes are the same as for SYSPRINT.

**SYSUDUMP DD STATEMENT** SYSUDUMP requests an abend dump if major errors occur (note that most internal abends in FATAR are for the user's information only and do not cause dumps). SYSUDUMP is usually allocated to SYSOUT. Abend dumps are necessary for analysis of problems by INNOVATION DATA PROCESSING.

CONTINUED . . .

## 23.0 CONTINUED . . .

**TAPEIN DD  
STATEMENT**

TAPEIN specifies the tape or tapes to be analyzed. One or more volumes can be specified. "LABEL=" can specify "SL", "AL", "NL", or "BLP"; FATAR's decision to process the tape as labeled or unlabeled is independent of JCL. However, the "LABEL=" parameter can be used to position the input tape to a particular file at which to start processing. If "SL" or "AL" is specified, the data set name of the first file to be processed must be properly specified, and FATAR will backspace to the header labels of that input file before starting. Since this is an input file, "DISP=OLD" is appropriate.

If labels are being processed from TAPEIN, FATAR will usually receive sufficient information from those labels to deblock the data blocks in each file. However, if label information is missing or incorrect, it can be supplied by the "DCB=" parameter on TAPEIN. Even if label information is present, FATAR can be forced to treat input blocks as single large records for printing and modification by putting "DCB=RECFM=U" on the TAPEIN statement (the DCB information in the labels will still be used to open TAPEOUT files).

If "LABEL=(,AL)" or "DCB=OPTCD=Q" appears on the DD statement, or if FATAR detects ANSI labels on the tape, all labels and data read will be translated from ASCII to EBCDIC.

EXAMPLES :

```
//TAPEIN DD DSN=FILE.NAME,UNIT=TAPE,
// VOL=SER=111111,DISP=OLD
//TAPEIN DD DSN=ANY.FILE,UNIT=TAPE,
// VOL=SER=(222222,333333),
// LABEL=(4,BLP),DISP=OLD
```

**TAPEOUT DD  
STATEMENT**

The TAPEOUT DD statement is optional, but if present specifies a tape on which to create a copy of the input files read from TAPEIN. The "LABEL=" parameter on the TAPEOUT DD statement may be anything (SL, AL, NL, BLP) and FATAR will create the specified type of labels. Note that "SUL" or "AUL" is not required for the creation of user labels on TAPEOUT; if the input tape contains user labels, they will be copied to TAPEOUT (if labeled) unless the "SUPULAB" parameter is specified. If labels are not being processed on TAPEIN, the "DSN=" JCL parameter for TAPEOUT will be used to name every file created on TAPEOUT, but if labels are processed on TAPEIN, the "DSN=" is ignored, and every file name will be copied from TAPEIN. Since this is an output file, "DISP=NEW" is usually appropriate. Volume serials may be specified, or may be omitted to call for scratch tapes.

The "LABEL=EXPDT=" or "LABEL=RETPD=" parameter may be specified to provide an expiration date for the output files created. If omitted, the expiration dates of labeled files read from TAPEIN will be used.

If "LABEL=(,AL)" or "DCB=OPTCD=Q" appears on the DD statement, all data written will be translated from EBCDIC to ASCII.

EXAMPLES :

```
//TAPEOUT DD DSN=NOT.USED,UNIT=TAPE,DISP=(,KEEP)
//TAPEOUT DD DSN=FILE.NAME,UNIT=TAPE,
// VOL=SER=444444,DISP=(,KEEP),DCB=DEN=3
```

**SYSIN DD  
STATEMENT**

The SYSIN DD statement is the source of FATAR control statements. It is normally a "DD \*" spool file, but can be any disk or tape file with DCB characteristics RECFM=FB and LRECL=80. It can also be omitted entirely, in which case FATAR will assume all defaults and no modification or scan statements.

CONTINUED . . .

**23.0 CONTINUED . . .**

**JCL WHEN  
FATAR IS A  
FATS SUB-  
TASK** FATAR may be invoked as a sub-task of its companion product FATS in order to execute more than one FATAR task simultaneously in the same jobstep. In this case several differences from the above rules apply (in each case, the "x" referred to is the number "x" in the FATS control statement "ANALYZE(x)"):

- FATAR will write its report to Print DD statement SYSPRINx rather than SYSPRINT (e.g., SYSPRIN1).
- The TAPESUMM DD statement is not supported.
- The TAPEx DD statement rather than TAPEIN will be used for the input tape (e.g., TAPE1).
- If present, the TAPExOUT DD statement rather than TAPEOUT will be used for the output tape copy.
- Most ANALYZE statement parameters can be specified in the FATS control statements. However, if you wish to override what FATS has specified, or wish to specify scan or modification control statements, use a SYSINx rather than a SYSIN DD statement (e.g., SYSIN1).

## 24.0 FATAR CONTROL STATEMENTS

### 24.1 GENERAL

**GENERAL RULES** All FATAR control statements must be input on 80-character records of which only columns 1 to 71 are used by FATAR.

Every FATAR control statement must begin with an operation keyword from the list below. This keyword may begin in column 1 or be preceded by any number of spaces. It must be followed by one or more spaces. Every operation keyword accepts a number of keyword parameters which follow those blanks and are separated by commas with no intervening spaces. Control statements may be continued onto multiple input records by following the last keyword parameter with a comma and a space, and starting the next keyword on the next record in any column (preceded by spaces).

Any FATAR control statement may contain comments; FATAR stops its scan of the control statement whenever it finds a keyword parameter followed by a space (or comma-space for continuation), so the rest of the statement may be filled with any comment desired. Any control statement record containing an asterisk ("\*") in column 1 will be bypassed by FATAR and may entirely contain comments.

FATAR operation keywords:

|         |   |
|---------|---|
| ANALYZE | Perform FATAR analyze/copy (required)   |
| DROP    | Do not copy one or more blocks  |
| KEEP    | Copy a block even if permanent data checks or change its length                                       |
| PRINT   | Print one or more blocks  |
| REPLACE | Replace data in a block or record   |
| SCAN    | Validate data fields in a block or record for packed or zoned numeric or against a user-defined table |
| TABLE   | Define valid characters for a scan operation  |

In the statement formats which follow, parameters enclosed in square brackets are optional; if there is a vertical list of parameters or optional formats in the brackets, only one may be chosen (defaults are underlined). A vertical list of parameters enclosed in braces indicates that one of the parameters or formats must be specified.

[ ] – Brackets–Option parameters

{ } – Braces–Required parameter  
Choose one

## 24.2 ANALYZE CONTROL STATEMENT

|         |  |                            |
|---------|--|----------------------------|
| ANALYZE | ,BLKSIZE=N                                 | ,METRIC                    |
|         | ,BLP                                       | , <u>NONMETRIC</u>         |
|         | , <u>CHAR</u>                              | ,MSGLEVEL=0                |
|         | ,DUMP                                      | ,NUMFILES                  |
|         | ,HEX                                       | ,NF =n                     |
|         | ,COMBFILES                                 | OUTBLP                     |
|         | ,CRDATE=COPY  <u>TODAY</u>                 | ,PRTLEN=n                  |
|         | ,ENDAFTER=(LF=n,B=b)                       | ,REBLOCK=n                 |
|         | ,EOV=NO                                    | ,MAXRBLK=m                 |
|         | ,EXIT=(OPENEXIT,exitname)                  | ,RETCODE                   |
|         | ,KEEP                                      | ,RETRY=n                   |
|         | ,LABELS= <u>YES</u>  NO OS                 | ,ROR=NO                    |
|         | ,LBLPRT= <u>CHAR</u>  HEX DUMP FORMAT NONE | ,STOP                      |
|         | ,LENCHK                                    | ,SUPULAB                   |
|         | ,LINECNT=n                                 | ,VALIDATE=INPUT OUTPUT ALL |
|         | ,MAXERR=n                                  | ,VERIFY                    |
|         | ,MAXTERR=n                                 | ,WTO                       |

CONTINUED . . .

**24.2 CONTINUED . . .**

The ANALYZE Control Statement is required (unless FATAR control statement input is omitted altogether). However, all of its keyword parameters are optional. Keywords may appear in any order on the statement. Keywords are available to control the input tape operation, error recovery actions and printing plus a few others; they are presented in those categories below. Only one ANALYZE control statement may be input.

If ANALYZE has no parameters, or if the control statement data set is omitted altogether, FATAR will perform the following functions:

- Input tape will be tested for presence of labels. If not labeled, all files on one tape will be read up to a double tape mark (end of tape). If labeled, all files on all input volumes will be read until a double tape mark is read or the list of volumes is exhausted (EOV labels will cause tape switching).
- If labels are ANSI, all labels and data will be translated from ASCII to EBCDIC.
- If present, all labels will be printed. If the "TAPESUMM" DD statement is present, a summary of the files read will be printed for each input tape.
- All temporary and permanent data checks will be noted, including printing 80 bytes from the blocks preceding and following the data check block.
- If DCB information is available (from tape labels or JCL), all blocks on TAPEIN will be checked to ensure that they don't exceed the blocksize, that they contain validly formatted fixed or variable records, and, for variable spanned records, that the record segments are valid.
- If the TAPEOUT DD statement is present, all data read from TAPEIN will be written to TAPEOUT, except that data check blocks will be dropped. The label type of TAPEOUT is controlled by the JCL for TAPEOUT. If TAPEIN is labeled, the output data sets will have the same names as the input data sets.

## 24.2.1 INPUT TAPE CONTROL KEYWORDS

- BLKSIZE=n** "n" specifies the maximum size block (in bytes) which will be accepted from the input tape. Blocks larger than "n" on the tape will cause FATAR to be terminated. Specifying "BLKSIZE=n" will reduce the memory requirements of FATAR by 65535-n (rounded to the next lower multiple of 4096). If the memory requirements of FATAR are not a problem, and if you have no need to scan for over-large blocks, do not specify "BLKSIZE=".
- "n" may be 4-65535  
Default: BLKSIZE=65535
- BLP** The "BLP" parameter is provided to help when installation conventions do not allow "LABEL=(,BLP)" to be specified on a JCL DD statement, but it is desired to bypass label processing on TAPEIN so that FATAR can open and process any tape (such as when the true volume serial is unknown). When "BLP" is specified on the "ANALYZE" statement, the JCL label parameter will be changed to "BLP" before the tape is opened. If a file number is specified in JCL it is not changed, e.g., "LABEL=3" will become "LABEL=(3,BLP)". "BLP" can be used under MVS systems only if FATS is installed and executed as an authorized program (see the installation instructions). "BLP" may still be disallowed by installation open exits.
- ENDAFTER=(LF=n,B=b)** The "ENDAFTER" parameter causes FATAR to stop reading after a certain block in a given file has been processed; it is much like the "NUMFILES=" parameter above except that it can terminate at any point within a file. "n" is the logical file number (see "MODIFICATION/SCAN parameters") and "b" is a decimal number giving the block number (the first block in each file is block 1).
- "n" may be 1-32767, "b" may be 1-999999.
- EOV=NO** Whenever labels are not being processed on the input tape, FATAR will still check for EOV labels when "NUMFILES=0" since EOV labels will not be followed by two tape marks, and may allow the tape to run off of the reel. Also, if copying to an output tape, block counts will be updated in any EOF or EOV labels encountered. Specifying "EOV=NO" suppresses both of these actions.
- LABELS=YES|NO|OS** "LABELS=YES" specifies that the input tape is to be checked for standard IBM labels or ISO/ANSI labels; if present, these labels will be processed and printed, and the data set name and DCB information will be used for copying the data set; if not present, the tape will be treated as unlabeled.
- "LABELS=NO" forces the input tape to be treated as unlabeled (even if it does have labels); all physical files will be treated as data.
- "LABELS=OS" specifies that the operating system is to handle all label processing; "NUMFILES=1" is assumed and JCL parameters control whether the tape is labeled or not.
- Default: LABELS=YES
- NOTE:** "LABELS=YES" will be automatically changed to "LABELS=NO" if the first block read is not a label; this means that "LABELS=" rarely needs to be used since FATAR will properly handle a labeled or unlabeled tape automatically.

CONTINUED . . .

## 24.2.1 CONTINUED . . .

- NUMFILES=n**  
**NF=n** "n" is a decimal number specifying the number of logical files which FATAR is to read from the input tape. A logical file is one standard label file if labels are being processed, or one physical file if not. If labels are processed and an input file extends to more than one volume, it is still considered one logical file. If "n" is 0, FATAR will read all files until two consecutive tape marks are read (signifying the logical end-of-tape), or until all input tape volumes have been processed; this is the default. Also, if labels are not being processed and "n" is 0, FATAR will stop if EOVL labels are read. Note that if "n" is a value greater than the actual number of files on a tape, the tape may run off of the reel and require operator intervention to rethread it. "n" may be 1-32767.
- NOTE:** 3480 tape cartridges cannot run off the end.  
Default: NUMFILES=0
- STOP** When "STOP" is specified, message FATS012 will be written to the operator at the end of each file read from TAPEIN, giving the option of terminating FATAR or continuing. This provides a measure of dynamic control of FATAR.
- WTO** When "WTO" is specified, message FATS011 will be written to the operator at the end of each file read from TAPEIN or verified from TAPEOUT, giving the block and error counts.

## 24.2.2 ERROR RECOVERY CONTROL KEYWORDS

- KEEP** By default, FATAR will not copy to the output tape any blocks which had permanent data checks. If the "KEEP" parameter is specified, FATAR will write to the output whatever data it was able to read during the last recovery attempt. Note that the block length may be incorrect due to the data check.
- MAJERR=n** If "n" major errors (FATS046, 7, 8, 9 messages) are detected, FATAR will issue message FATS016 and abend with a U0888 Abend. "n" may be 1-32767.  
Default: MAJERR=1
- MAXERR=n** If "n" permanent data checks are detected, FATAR will abend with a U0888 Abend. "n" may be 1-32767.  
Default: MAXERR=20
- MAXTERR=n** If "n" temporary data checks are detected, FATAR will abend with a U0888 Abend. A temporary data check is one which FATAR reads successfully during recovery processing. "n" may be 1-32767.  
Default: MAXTERR=100
- RETRY=n** "n" specifies the number of times FATAR will attempt to re-read a block causing a data check before considering the error permanent. FATAR will attempt this many retries in each direction (forward read and backward read). "n" must be from 0 to 256. "RETRY=0" suppresses all re-read attempts and considers the error permanent immediately; if a tape drive cannot seem to "get past" an error on a tape, this may allow the tape to be read. For IBM 3480 tape cartridge systems, because of the extensive error recovery done by the tape control unit, "RETRY=0" will be forced.  
Default: RETRY=40 (RETRY=0 on 3480)
- ROR=NO** "ROR=NO" suppresses "READ-OPPOSITE-RECOVERY", the attempt to recover from a data check by reading it in the backwards direction. With this parameter specified, FATAR will only attempt forward re-reads for data checks. "ROR=NO" is assumed for 7-track tape with data conversion on.

## 24.2.3 OUTPUT TAPE CONTROL KEYWORDS

- CRDATE=COPY|TODAY** If TAPEOUT is a labeled tape, the creation date recorded in the labels on TAPEOUT will be the current run date, unless "CRDATE=COPY" is specified, when the creation date will be copied from the corresponding labels on TAPEIN if it is also labeled. CRDATE=COPY is not recommended for CA1 (UCC-1/TMS) tape management system users since it affects the expiration date on TAPEOUT in unpredictable ways.  
Default: CRDATE=COPY
- OUTBLP** The "OUTBLP" parameter is provided to help when installation conventions do not allow "LABEL=(,BLP)" to be specified on JCL DD statement, but it is desired to bypass label processing on TAPEOUT so that FATAR can open and process any tape (such as when the true volume serial is to be overwritten in an image copy). When "OUTBLP" is specified on the "ANALYZE" statement and "DEFER" is coded in the UNIT parameter of the DD statement, the JCL LABEL parameter will be changed to "BLP" before the tape is opened. If a file number is specified in JCL it is not changed, e.g., "LABEL=3" will become "LABEL=(3,BLP)". "OUTBLP" can be used under MVS systems only if FATS is installed and executed as an authorized program (see the installation instructions). "OUTBLP" may still be disallowed by installation open exits.
- MAXRBLK=n** "n" is a decimal number giving the largest current blocksize to be considered for reblocking ("MAXRBLK=" is meaningful only if "REBLOCK=" is also specified). Any input file whose existing blocksize is greater than "n" will not be reblocked. "n" may be 4-32767.  
Default: All files are reblocked (if "REBLOCK=" given).
- REBLOCK=n** "n" is a decimal number specifying the target blocksize for data reblocking. When this keyword is specified, all files will be reblocked as they are copied to have blocksizes as close to "n" as possible. "n" may be 4-32767.  
Default: Blocks are copied as read from TAPEIN.
- SUPULAB** If any file read from the input tape contains user labels and the output tape is a labeled tape, the user labels will be copied to the output tape unless "SUPULAB" is specified.  
Default: User labels copied if output tape labeled

## 24.2.4 PRINT CONTROL KEYWORDS

**CHAR|HEX|DUMP**

"CHAR" specifies that FATAR will print any required blocks or records in EBCDIC characters, 80 per print line. "HEX" specifies that FATAR will print in hexadecimal format, 8 groups of 4 bytes (32 bytes total) per print line. "DUMP" specifies a combination, with the 32 characters of EBCDIC printed to the right of the same 32 bytes of data in hexadecimal. This format can be overridden for specific blocks on a "PRINT" Control Statement.

Default: CHAR

**LBLPRT=CHAR|HEX|DUMP|  
FORMAT|NONE**

When labels are being processed on the input tape, FATAR will normally print those labels. By default, labels are printed in EBCDIC character format regardless of the "HEX" or "DUMP" parameters described above. "LBLPRT=" specifies an overriding print format for labels only. The values "CHAR", "HEX", and "DUMP" have the same meanings as the equivalent parameters described above. "LBLPRT=FORMAT" causes the labels to print in character format, followed by a two line breakdown of the individual fields in each label so that the contents of those fields can be determined more easily. "LBLPRT=NONE" suppresses label printing.

Default: LBLPRT=CHAR

**LINECNT=n**

"n" specifies the number of lines per page to print on all FATAR print data sets. "n" may be 0-32767.

Default: LINECNT=60

**MSGLEVEL=0**

By default, FATAR will print the contents of any block which has been modified by a "REPLACE" or "KEEP" Statement (up to the limit specified by "PRTLEN="). "MSGLEVEL=0" will suppress this print (FATAR will continue to print a line documenting that the block was modified).

**PRTLEN=n**

Whenever a block from the input tape is to be printed, for whatever reason, FATAR will print only the first "n" bytes or characters of data unless overridden by a "PRINT" Control Statement. If the block is being deblocked, FATAR will print as many logical records as exist in the first "n" bytes (and may truncate the last record). "n" may be 0-32767.

Default: PRTLEN=80

**NONMETRIC  
METRIC**

Applies to: ALL

When NONMETRIC is specified (or defaulted) FATAR will list tape lengths and error positions and lengths in American units. When METRIC is specified FATAR will list tape lengths and error positions and lengths in metric units.

Default: NONMETRIC

## 24.2.5 MISCELLANEOUS KEYWORDS

|                                  |   |
|----------------------------------|---|
| <b>COMBFILES</b>                 | Combine all files found on TAPEIN into a single file on TAPEOUT. You must be aware of any changes in DCB characteristics from file to file and your application must be able to handle those changes when it reads the tape you are creating. If you are using FATAR PRINT, REPLACE, SCAN, or DROP statements and the DCB characteristics of one of the files is different from the first file, you may experience additional FATAR error messages. If you use the LENCHK parameter you will likely fill up your spool with error messages if one or more of the files differs from the characteristics of the first file.  |
| <b>EXIT=(OPENEXIT,EXITNAME)</b>  | Activate FATAR user open exit facility described in <a href="#">Section 21.4</a> , output tape processing. The parameter "EXITNAME" is the name of the load module to be loaded by FATAR and executed at open time for TAPEOUT.<br><br>Default: No exit is activated.   |
| <b>LENCHK</b>                    | If "LENCHK" is specified, FATAR will compare the length of every block read to the length of the preceding block and print the contents of the block if the lengths do not match. The first block in every file will always print. This can be used to scan a fixed length file for improper length blocks.   |
| <b>RETCODE</b>                   | "RETCODE" causes FATAR to terminate with a return code (condition code) of 12 rather than a U0200 or U0888 abend when serious errors are found.   |
| <b>VALIDATE=INPUT OUTPUT ALL</b> | If "LABEL=(,BLP)" is specified on TAPEIN or TAPEOUT DD Statements, data management will not VERIFY the volume serial or expiration date of the tapes. However, if "VALIDATE=" is specified, FATAR will VERIFY the volume serial of TAPEIN ("INPUT"), TAPEOUT ("OUTPUT") or both ("ALL)". The volume serial compared is the serial in the DD statement. For TAPEOUT, the expiration date in the tape label will be checked. If the wrong volume is mounted or if the expiration date is not yet reached, FATAR will issue message FATSW08 to the system operator giving the option of ignoring the error, mounting another tape, or terminating. TAPEOUT validation applies only to the first output volume.<br><br>Default: No validation |
| <b>VERIFY</b>                    | If "VERIFY" is specified and an output tape is being produced, at the conclusion of the copy FATAR will reopen the output tape as input and re-read it to VERIFY its readability and contents (it is not compared to the input tape). All FATAR options are reset to the DEFAULT and all MODIFICATION and SCAN Control Statements are ignored during the verification.  |

### 24.3 MODIFICATION/SCAN CONTROL STATEMENTS

#### MODIFICATION & SCAN STATEMENTS

The FATAR block MODIFICATION and SCAN Control Statements (PRINT, DROP, KEEP, REPLACE, and SCAN) invoke optional facilities of FATAR to print data from TAPEIN, to scan TAPEIN data blocks for certain data or types of data, and to modify TAPEIN data blocks before they are written to TAPEOUT.

There may be as many MODIFICATION/SCAN Statements as required. More than one statement may refer to the same block on TAPEIN; if so, the statements are executed in the order in which they appear in the control statement input. An exception to this is the PRINT Statement, since each data block will be printed only once; the last PRINT Statement which applies to a given data block is the one which will control its printing.

All parameters must be specified on the MODIFICATION/SCAN Control Statements in the order shown in the statement formats below. However, it is not necessary to indicate the absence of optional parameters with a comma.

#### 24.3.1 PRINT STATEMENT

##### PRINT STATEMENT

**PRINT** LF=ALL|nH|n|nT ,B=ALL|b<sub>1</sub>|b<sub>1</sub>-b<sub>2</sub> ,L=l ,CHAR ,HEX ,DUMP

##### PRINT ALL

The PRINT Statement requests that data blocks from TAPEIN be printed after other MODIFICATION Statements have been applied. By default, the length and format of the printout are controlled by the "PRTLEN=", "CHAR", "HEX", and "DUMP" parameters on the ANALYZE Statement (which themselves default to 80 bytes in character format), but this can be overridden by the "L=" and "CHAR/HEX/DUMP" parameters on the PRINT Statement itself.

Note that the Print length refers to the number of characters or bytes to print from each block, even if the blocks are being deblocked into logical records. If the length to print is longer than one logical record, the printout will be formatted into logical records.

The "PRINT ALL" Statement causes all blocks in all files on TAPEIN to be printed, and cannot have any other parameters on it. In this case, the length and format are controlled by the above-mentioned ANALYZE parameters.

#### 24.3.2 DROP STATEMENT

##### DROP STATEMENT

**DROP** DCK, LF=ALL|nH|n|nT ,B=ALL|b<sub>1</sub>|b<sub>1</sub>-b<sub>2</sub>|b<sub>1</sub>.r<sub>1</sub>|b<sub>1</sub>.r<sub>1</sub>-b<sub>2</sub>.r<sub>2</sub>

The DROP Statement causes one or more blocks or records to be dropped (not copied from TAPEIN to TAPEOUT). If the "DCK," parameter is specified, the drop will be done only if the affected block had a permanent data check when read from TAPEIN. If "B=ALL", the entire file referenced by "LF=" will be dropped from TAPEOUT, including all tapemarks and labels (to drop all data from a file without dropping the file itself, specify "B=1-999999").

Dropping individual records (the "B=b<sub>1</sub>.r<sub>1</sub>" or "B=b<sub>1</sub>.r<sub>1</sub>-b<sub>2</sub>.r<sub>2</sub>" formats), is effective only if the blocks on TAPEIN are being deblocked into records. If all of the records in a block are individually dropped, the entire block will be dropped.

## 24.3.3 KEEP STATEMENT

**KEEP STATEMENT**    **KEEP DCK,**    **LF=ALL|nH|n|nT**    **,B=ALL|b1|b1-b2**    **,L=l,P=p,J=R**

The KEEP Statement causes blocks from TAPEIN to be copied to TAPEOUT, even if they would be dropped because of a DROP Statement, or because of a permanent data check (the "DCK," parameter is required to keep a data check block).

If the optional "L=" parameter is given, the KEEP Statement may be used to change the length of a block as it is being copied from TAPEIN to TAPEOUT. "L=" specifies the new block length. "P=", if given, specifies location in the new block (relative to 1) where the old data will be placed. "J=R", if given, requests that the old data be right-justified in the new block, with the rightmost byte of the old data block positioned at the location specified by "P=". These combinations allow full control over repositioning of the data. Any extra bytes added at the front or end of the new block will be set to spaces (blanks). Whenever the length of a block is changed, deblocking of that block into logical records will be terminated.

## 24.3.4 SCAN STATEMENT

**SCAN STATEMENT**    **SCAN DCK,**    **LF=ALL|nH|n|nT**    **,B=ALL|b1|b1-b2|b1.r1|b1.r1-b2.r2**    **,L=l,P=p,T=t**

The SCAN Statement causes specified data fields to be validated for specific type of content. Scans are performed on a logical record basis if records are being deblocked. The field to be scanned is at the location specified by "P=" (relative to 1) for the length specified by "L=". If "T=Z" is specified, the field is validated for valid zoned decimal characters including decimal sign characters. If "T=P" is specified, the field is validated for valid packed decimal characters. For any other value of "T=", there must be a corresponding TABLE Statement with the same "T=" value, and the SCAN Statement will validate the presence or absence of the characters specified there. A message will be issued for every field which fails the SCAN test.

## 24.3.5 REPLACE STATEMENT

**REPLACE STATEMENT**      **REPLACE DCK,**                      **LF=ALL|nH|n|nT**                      **,B=ALL|b1|b1-b2|b1.r1|b1.r1-b2.r2**                      **P=p**  
**,V|,S=X'HEX'|C'EBCDIC'**                      **,D=X'HEX'|C'EBCDIC'**                      **,ALL**

The REPLACE Statement is used to modify data within blocks being copied from TAPEIN to TAPEOUT, and also to scan for blocks or records containing a specified data string (even if TAPEOUT is not present). If TAPEIN data blocks are being deblocked, the REPLACE Statement operates on a logical record basis.

If "D=" is specified without "V=" or "S=", the data string specified by "D=" will be placed in every affected data record or block at the location given by "P=".

If "D=" is specified with "V=", the data in every affected data record or block will be verified for the current contents specified by "V=" before the data from "D=" is placed there (again, the location within the block or record is given by "P="). This allows you to be sure the proper data field is being replaced, or to search a set of records for the proper one to modify.

If "D=" is specified with "S=", every affected record or block will be scanned (starting at the location specified by "P=", default 1) for the data string specified by "S=". If found, the replacement data given by "D=" will be stored there. Normally, only one such replacement will be done in each record or block, but if the parameter "ALL" is also given, the scan will continue until the end of the record/block.

If "V=" or "S=" is given without "D=", verification or scanning will take place as described above, but no data modification will be done. This allows the REPLACE Statement to be used to scan data for specific data strings, since any block on which REPLACE successfully operates will be eligible for printing.

## 24.3.6 TABLE STATEMENT

**TABLE STATEMENT**      **TABLE T=t ,R**                      **,D=X'HEX'|C'EBCDIC'**

The TABLE Statement is used to define a set of characters for SCAN Statement operations. "T=" must specify a unique letter (except for P and Z which are reserved) for naming the table, so up to 24 tables can be defined. Each table can contain up to 256 characters. The TABLE Statement must precede any SCAN Statement which references it. If a field being scanned contains any character not in the table, it will be flagged and its block printed. If the optional parameter "R" is included, this operation is reversed, so that if the field contains any character which is in the table, it is flagged.

## 24.4 MODIFICATION/SCAN PARAMETERS

The following are details of the keyword parameters which may appear on the MODIFICATION/SCAN Statements above. Consult the definitions of each statement to see which parameters are valid on each statement, and in what form.

|  |   |
|--|---|
| <b>ALL</b>   | On a REPLACE statement containing the "S=" parameter, specifies that all occurrences of the scan string in the current block or record are to be replaced. If omitted, only the first occurrence of the scan string is replaced.  |
| <b>B=ALL b<sub>1</sub> b<sub>1</sub>-b<sub>2</sub> b<sub>1</sub>.r<sub>1</sub> b<sub>1</sub>.r<sub>1</sub>-b<sub>2</sub>.r<sub>2</sub></b> | Specifies to which blocks within the file specified by "LF=" the statement applies. "B=ALL" requests all blocks in the file. "B=b <sub>1</sub> " causes it to apply only to one specified block. "B=b <sub>1</sub> -b <sub>2</sub> " specifies a range of blocks ("b <sub>1</sub> " less than "b <sub>2</sub> "). "B=b <sub>1</sub> .r <sub>1</sub> " indicates a specific record within a specific block (record deblocking must be in effect). "B=b <sub>1</sub> .r <sub>1</sub> -b <sub>2</sub> .r <sub>2</sub> " causes the statement to apply to a range of records ("b <sub>1</sub> " and "b <sub>2</sub> " do not need to be the same). "b <sub>1</sub> ", "b <sub>2</sub> ", "r <sub>1</sub> ", and "r <sub>2</sub> " are all unsigned decimal numbers. Blocks within a file and records within a block are numbered starting with 1.   |
| <b>CHAR HEX DUMP</b>   | On a PRINT Statement, specifies an overriding print format (character, hexadecimal, or dump) for this PRINT Statement only. If omitted, the print format specified on the ANALYZE Statement will control the format for this PRINT Statement.   |
| <b>DCK</b>   | Specifies that FATAR is to perform the indicated operation only if the block affected had a permanent data check when read from TAPEIN.   |
| <b>J=R</b>   | On KEEP Statements only, specifies that the old data block is to be right-justified in the new data block (if not specified, the old data block is left-justified).   |
| <b>L=l</b>   | For PRINT, specifies an overriding print length.<br>For KEEP, specifies the new length of the block.<br>For SCAN, specifies the length of the field to verify. "l" must be an unsigned decimal number.  |
| <b>LF=ALL nH n nT</b>  | Specifies the logical file on the input tape to which this statement applies. "n" is a decimal number from 1 to 32767 giving the logical file number. If the input tape is labeled, this is the file sequence number, the same value you would specify in the "LABEL=n" parameter in JCL. Specifying "n" alone will cause the statement to apply to the equivalent data file. Appending an "H" or "T" after the file number causes it to apply to the header or trailer label file, respectively, preceding or following that data file. If the input tape is not labeled, or if labels are not being processed ("LABELS=NO"), then "n" is the physical file number relative to the beginning of the tape (file 1).<br><br>LF=ALL specifies that this statement applies to all files of the input tape.<br><br><b>NOTE:</b> For compatibility with Version 3.0 of FATAR, the parameter "F=n" will be accepted in place of "LF="." However, in this case, "n" specifies the physical file number only regardless of the label status of the tape, and header and trailer label files must be counted. Use of "F=" is not recommended for new applications. |

CONTINUED . . .

## 24.4 CONTINUED . . .

**P=p** For SCAN and REPLACE, specifies the starting location for the operation (if FATAR is deblocking logical records, specifies a location within a record). For KEEP, specifies the starting location (or ending location if "J=R") of the old data block within the new block.

"p" must be an unsigned decimal number. "P=1" is the default and specifies the first byte of the record or block. For deblocked variable format records, the first data byte is at location 5.

**T=t** On a TABLE Statement, defines the name of a table for reference by a SCAN Statement. "t" must be a unique letter except P and Z.

On a SCAN Statement, specifies the type of scan to be performed:

T=Z – scan for zoned decimal numbers

T=P – scan for packed decimal numbers

T=t – scan according to user table "t"

**V**  
**S=X'HEX'|C'EBCDIC'**  
**D**

Defines a data string to FATAR. The data may be in characters (C) or hexadecimal (X). For character data, each character defines one byte of data, and may be any EBCDIC character. If a quote must be part of the data, specify it as two consecutive quotes (e.g., C 'AB' 'C"). In hexadecimal data, each pair of hex digits defines one data byte, so the total number of digits must be even. Hex digits may be the numbers 0-9 and the letters A-F. For readability, hex input may be broken up by commas (e.g., x '09A6,F1,F2'). A maximum of 256 bytes may be specified in one data string.

A data string may be continued on the same card by closing the data with a quote, followed by a comma, followed by a X' or C' and more data. In this way, a string may contain mixed character and hexadecimal data, up to the maximum of 256 bytes. Similarly, a data string is continued onto another card by closing the string with a quote and comma followed by a blank, and starting on the next card (in any column) with X' or C'.





**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## 26.0 FATAR EXAMPLES OF USAGE

### **INTRO- DUCTION**

Examples are provided here to guide you in the use of FATAR. As many of the common uses of FATAR as possible have been included, and most of the control statements and keywords are illustrated here. However, be aware that they are just examples and must be customized to your installation and intended use. Areas of customization will probably include: data set names, unit names for tape, special parameters for tape management systems. The examples also assume that no special "STEPLIB" DD Statement is required to execute FATAR; this may not be true in your installation.

Because of the many possible combinations of options, control statements, tape label types, etc., it is not practical to create examples which cover all possible FATAR uses. Many examples illustrate more than one aspect of FATAR use, e.g., a particular tape label type and use of some FATAR keyword. This does not mean that the two must always go together unless the accompanying text says so.

Some examples use "LABEL=(,BLP)" on the input tape in order to be able to mount a labeled tape without knowing its volume serial or data set names. Sometimes this is simply a convenience, but for some operations, such as printing the labels from an unknown tape, it is essential. In some installations, system parameters prevent the usage of "BLP" by most users; in this case, the FATAR option "BLP" on the "ANALYZE" Statement can be substituted.

## 26.1 INDEX TO EXAMPLES

**INDEX TO  
EXAMPLES**

The first-time or infrequent user of FATAR should review the examples below in order since they frequently build upon one another in illustrating the usage and effects of various parameters. For your convenience, however, here is a list of the examples provided to aid in quickly finding the one you need. It shows the title of the example, the "ANALYZE" Statement keywords it illustrates, and the other Control Statements it illustrates.

| #  | <u>EXAMPLE NAME</u>   | <u>KEYWORDS</u>                  | <u>STATEMENTS</u>    |
|----|---|----------------------------------|----------------------|
| 1  | Analyze any tape  |                                  |                      |
| 2  | Analyze an SL tape  | LENCHK,LBLPRT                    | PRINT                |
| 3  | Analyze a partial SL tape                                   | NUMFILES                         | PRINT ALL            |
| 4  | Copy an SL tape   |                                  |                      |
| 5  | Copy any tape to an SL tape                                 | VERIFY,KEEP                      | DROP                 |
| 6  | Partial copy of an SL tape                                  | NUMFILES,REBLOCK                 |                      |
| 7  | Partial copy of an SL tape in BLP mode                      |                                  | PRINT                |
| 8  | Analyze a multi-volume tape set                             |                                  | PRINT,SCAN           |
| 9  | Analyze an SL tape with no label processing                 | LABELS,RETRY                     | PRINT                |
| 10 | Analyze an NL tape  | NF,PRTLEN                        | PRINT,SCAN,<br>TABLE |
| 11 | Search a partial file                                       | ENDAFTER, PRTLEN,<br>LBLPRT      | REPLACE              |
| 12 | Copy an ANSI tape   |                                  |                      |
| 13 | Copy an NL tape   | NUMFILES, VERIFY                 | KEEP                 |
| 14 | Copy an SL file to NL                                       | NUMFILES                         |                      |
| 15 | Image copy an SL tape                                       | LABELS                           |                      |
| 16 | Copy a multi-file, multi-volume tape<br>to a 3480 cartridge | VERIFY                           | REPLACE              |
| 17 | Copy and modify a tape file                                 | LABELS, MSGLEVEL                 | REPLACE              |
| 18 | Copy and modify a tape file                                 |                                  | REPLACE              |
| 19 | Recover from data checks                                    | RETRY,KEEP                       | REPLACE              |
| 20 | Recover from data checks on multi-<br>volume tape           |                                  |                      |
| 21 | Recover from data checks                                    | NUMFILES, LABELS                 |                      |
| 22 | Recover data from an overwritten tape                       | NF                               |                      |
| 23 | Copy files from an overwritten tape                         |                                  |                      |
| 24 | Copy an improperly closed tape                              | ENDAFTER                         |                      |
| 25 | Exclude files from a multi-file copy                        | VERIFY                           | DROP                 |
| 26 | Copy from 7-track input                                     |                                  |                      |
| 27 | Copy user labels  |                                  |                      |
| 28 | Drop an extra tape mark                                     | NUMFILES, LABELS                 | DROP                 |
| 29 | FATAR as a FATS sub-task – copy to 3480                     | VOL                              |                      |
| 30 | FATAR as a FATS sub-task – multiple tape<br>and drives      | VOL, VOLINCR, MAXVOLN,<br>VERIFY |                      |
| 31 | Copy and recatalog multiple files to<br>3490E format        | LABELS=OS,NUMFILES               |                      |

## 26.2 EXAMPLES

**EXAMPLE 1:** Analyze any tape. Since "LABEL=(,BLP)" is specified on the TAPEIN DD Statement, the tape may have any type of labels (IBM standard, ANSI, or none at all); FATAR will automatically determine the label type and handle them if present. If the labels are ANSI, both labels and data will be translated to EBCDIC. The presence of the optional TAPESUMM DD Statement will cause FATAR to print a summary of the files and data on the tape. FATAR will print all tape labels, count the blocks in every file, and identify any block causing a temporary or permanent data check, printing the first 80 characters of that block as well as the blocks preceding and following it. All blocks will be deblocked (if the tape is labeled). All files on the tape will be processed.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS ,UNIT=TAPE ,VOL=SER=999999 ,
//           LABEL=( ,BLP) ,DISP=OLD
//SYSIN      DD    *
              ANALYZE
```

**EXAMPLE 2:** Analyze standard label tape 111111. Since it is being opened as a labeled tape (no "LABEL" parameter on the TAPEIN DD Statement), the first data set name must be specified correctly. The parameter "LENCHK" causes FATAR to print the beginning of any block which differs in length from the preceding block (for checking fixed length files). The parameter "LBLPRT=FORMAT" causes labels to be printed field-by-field for easier analysis. The "PRINT" statement causes blocks 1 to 3 of every data file to be printed. All other operations are the same as Example 1 above.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS ,UNIT=TAPE ,VOL=SER=111111 ,
//           DISP=OLD
//SYSIN      DD    *
              ANALYZE LENCHK ,LBLPRT=FORMAT
              PRINT  LF=ALL ,B=1-3
```

**EXAMPLE 3:** Analyze the fourth and fifth files on standard labeled tape 111111, where the fourth file is called "FOURTHDS". The "LABEL=4" parameter on the TAPEIN DD Statement tells FATAR to begin with that data set, and the "NUMFILES=2" parameter on the "ANALYZE" Statement limits the number of files FATAR will analyze. The "PRINT ALL" Control Statement directs FATAR to print all data blocks (first 80 bytes).

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=FOURTHDS ,UNIT=TAPE ,VOL=SER=111111 ,
//           LABEL=4 ,DISP=OLD
//SYSIN      DD    *
              ANALYZE NUMFILES=2
              PRINT  ALL
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 4:** Standard labeled tape 111111 will be copied to standard labeled tape 222222. All files will be copied. Each file on tape 222222 will receive the data set name of the original file on tape 111111; the data set name on the TAPEOUT DD statement will not be used. If any blocks have unrecoverable data checks, they will be dropped from the output tape, and the block counts in the data set trailer labels adjusted.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS , UNIT=TAPE , VOL=SER=111111 ,
//           DISP=OLD
//TAPEOUT    DD    DSN=NOT . USED , UNIT=TAPE , VOL=SER=222222 ,
//           DISP=(NEW , KEEP )
//SYSIN      DD    *
ANALYZE
```

**EXAMPLE 5:** Tape 999999 will be copied to a scratch standard label volume. All files will be copied. FATAR will determine the label type of the input tape (STANDARD, ANSI, or UNLABELED) and process it appropriately. If the input labels are ANSI, all data files will be translated from ASCII to EBCDIC (if the records are ANSI variable length, RECFM=D, they will be converted to IBM RECFM=V). The data set names from the input tape will be copied to the output (unless TAPEIN is unlabeled, causing the TAPEOUT dsname to be used). The volume serial of the output tape will be preserved. If the output tape is filled before the input data is exhausted another volume will be requested. If any blocks on the input have unrecoverable data checks, they will be identified and printed, but the "KEEP" parameter will cause them to be written to the output tape anyway (using whatever data the input tape was able to transfer). The "DROP" statements will prevent the indicated input blocks from being written to the output tape. The "VERIFY" parameter will cause the output tapes to be analyzed after the copy is complete to verify readability.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS , UNIT=TAPE , VOL=SER=999999 ,
//           LABEL=( , BLP ) , DISP=OLD
//TAPEOUT    DD    DSN=NOT . USED , UNIT=TAPE , DISP=(NEW , KEEP )
//SYSIN      DD    *
ANALYZE KEEP , VERIFY
DROP  LF=3 , B=2
DROP  LF=3 , B=10 - 12
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 6:  
PARTIAL COPY  
OF AN SL TAPE**

Files 3 and 4 will be copied from labeled tape 111111 and written to a scratch tape. The "LABEL=3" parameter on the TAPEIN DD Statement causes FATAR to start the copy with file 3 of the tape; the data set name of that file must be properly specified. The "NUMFILES=2" parameter tells FATAR to copy two logical files, namely files 3 and 4. The files will become files 1 and 2 on the output tape, with the data set names of the original files. The "DCB=DEN=3" on the TAPEOUT DD statement will cause that tape to be written at 1600 BPI. "REBLOCK=16000" will cause each copied file to be reblocked to as close to a blocksize of 16000 bytes as possible.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=THIRDDS , UNIT=TAPE , VOL=SER=111111 ,
//           LABEL=3 , DISP=OLD
//TAPEOUT    DD    DSN=NOT . USED , UNIT=TAPE , DISP=(NEW , KEEP) ,
//           DCB=DEN=3
//SYSIN      DD    *
ANALYZE NUMFILES=2 , REBLOCK=16000
```

**EXAMPLE 7:  
PARTIAL COPY  
OF AN SL TAPE**

All files on labeled tape 111111 from file 4 through the end of the tape are copied to labeled output tape 222222, starting at file 2. In other words, file 4 will be copied to file 2, file 5 will be copied to file 3, etc. Data set names from the input tape will be retained. "DCB=RECFM=U" on the input tape will cause FATAR to treat the blocks as unblocked for printing/modification purposes but will not affect the DCB which will be written in the output tape's labels (which will be copied from the input tape's labels). "LABEL=(10,BLP)" is required on TAPEIN since we do not know its data set name and since FATAR must be positioned to the header labels of the first file. The rule is "SL file number, times 3, minus 2", in this case  $4*3-2=10$ .

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=UNKNOWN , UNIT=TAPE , VOL=SER=111111 ,
//           LABEL=( 10 , BLP ) , DISP=OLD , DCB=RECFM=U
//TAPEOUT    DD    DSN=NOT . USED , UNIT=TAPE , VOL=SER=222222 ,
//           LABEL=2 , DISP=(NEW , KEEP)
//SYSIN      DD    *
ANALYZE
PRINT ALL
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 8:** Analyze multi-volume labeled tape set 333333, 444444, and 555555, which contain more than one data set. FATAR will properly read all three volumes, giving total block counts for each data set even when it crosses volumes (an intermediate block count is given each time an end-of-volume is reached). The tape summary (DD statement "TAPESUMM") will contain one page for each tape, detailing the files and block counts on it. File numbers will be logical files since the beginning of the first tape, counting files which cross multiple volumes as one file. The "PRINT" control statements instruct FATAR to PRINT block 7 of file 1 in hexadecimal, blocks 22 and 23 of file 2 in dump format (HEX and EBCDIC) printing 96 bytes, and block 125 of file 4 in EBCDIC printing 250 bytes. The "SCAN" control statement requests FATAR to verify that the first 4 bytes of every record in file 2 contains a valid zoned decimal number.

```
//FATAR      EXEC   PGM=FATAR
//SYSPRINT   DD     SYSOUT=A
//TAPESUMM   DD     SYSOUT=A
//SYSUDUMP   DD     SYSOUT=A
//TAPEIN     DD     DSN=UNKNOWN,UNIT=TAPE,
//           VOL=SER=(333333,444444,555555),
//           LABEL=(,BLP),DISP=OLD
//SYSIN      DD     *
ANALYZE
PRINT  LF=1,B=7,HEX
PRINT  LF=2,B=22-23,L=96,DUMP
PRINT  LF=4,B=125,L=250
SCAN  LF=2,B=ALL,L=4,P=1,T=Z
```

**EXAMPLE 9:** Analyze SL tape 111111 without label processing, so that labels will be processed as data. Processing will continue until two consecutive tape marks are read or until "EOV" labels are read (tapes containing data sets which overflowed to another tape and end with "EOV" labels will not end with two tape marks, so FATAR has a special SCAN for the EOV labels). Since labels are not processed and no DCB information appears in the JCL, FATAR will process all tape blocks as "RECFM=U" (unblocked). "RETRY=5" instructs FATAR to limit its data check recovery attempts to 5 in the forward direction, and 5 in the backwards direction. Since FATAR is not processing labels, logical file numbers refer to the physical file number relative to the beginning of the tape, and the header and trailer label files are counted. The "PRINT" Statements cause the label records before and after the first data file on the tape to be printed in DUMP format.

```
//FATAR      EXEC   PGM=FATAR
//SYSPRINT   DD     SYSOUT=A
//TAPESUMM   DD     SYSOUT=A
//SYSUDUMP   DD     SYSOUT=A
//TAPEIN     DD     DSN=FIRSTDS,UNIT=TAPE,VOL=SER=111111,
//           DISP=OLD
//SYSIN      DD     *
ANALYZE LABELS=NO,
RETRY=5
PRINT  LF=1,B=ALL,DUMP
PRINT  LF=3,B=ALL,DUMP
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 10:** Analyze unlabeled tape 666666 with deblocking. The "NF=3" parameter limits FATAR to reading three physical files. Since DCB information is supplied on the TAPEIN DD statement, FATAR will attempt to deblock every file processed. On the "PRINT" and "SCAN" statements, the "LF=" parameter refers to the physical file number since the beginning of the tape. The "SCAN" statement is used to VERIFY that bytes 4 to 6 of all records from record 5 in block 3 to record 2 in block 5 contain only characters "A", "B", or "C" (as defined in the "TABLE" statement). The "PRTLEN=5000" will ensure that any blocks printed (due to the PRINT statement or failure of the SCAN statement) will print the entire block (up to 5000 bytes), deblocked into records.

```
//FATAR      EXEC   PGM=FATAR
//SYSPRINT   DD     SYSOUT=A
//TAPESUMM   DD     SYSOUT=A
//SYSUDUMP   DD     SYSOUT=A
//TAPEIN     DD     DSN=NOLABEL,UNIT=TAPE,VOL=SER=666666,
//           DCB=(RECFM=FB,LRECL=120),
//           LABEL=(,NL),DISP=OLD
//SYSIN      DD     *
ANALYZE NF=3,PRTLEN=5000
PRINT  LF=1,B=1-20
TABLE  T=A,D=C'ABC'
SCAN   LF=2,B=3.5-5.2,L=3,P=4,T=A
```

**EXAMPLE 11:** Analyze a partial file on SL tape 111111 and print blocks containing a certain string. Since only the first 20 blocks are to be scanned, there is no need to read the rest of the tape; the "ENDAFTER" parameter causes FATAR to terminate immediately after processing block 20 in file 1 (since labels are being processed, this is the first data file). The "REPLACE" statement tells FATAR to print any block containing the string "EQUIPMENT" in any position, and the "PRTLEN=32767" parameter causes the entire data block to be printed. The "LBLPRT=HEX" parameter causes the header labels of the file to be printed in hexadecimal.

```
//FATAR      EXEC   PGM=FATAR
//SYSPRINT   DD     SYSOUT=A
//TAPESUMM   DD     SYSOUT=A
//SYSUDUMP   DD     SYSOUT=A
//TAPEIN     DD     DSN=UNKNOWN,UNIT=TAPE,VOL=SER=111111,
//           LABEL=(,BLP),DISP=OLD
//SYSIN      DD     *
ANALYZE ENDAFTER=(LF=1,B=20),PRTLEN=32767,
           LBLPRT=HEX
REPLACE LF=1,B=1-20,S=C'EQUIPMENT'
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 12:** ANSI labeled tape 333333 is to be copied to a scratch tape with ANSI labels. The TAPEIN DD statement specifies "LABEL=(,AL)", but it could also specify "LABEL=(,BLP)" if the true volume serial or first data set name were not known; FATAR works correctly in either case. The TAPEOUT DD statement must specify "LABEL=(,AL)" for the labels to be written in ANSI format and the data in ASCII character set. All other processing is just the same as for standard label tapes (see example 4).

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS , UNIT=TAPE , VOL=SER=333333 ,
//           LABEL=( , AL ) , DISP=OLD
//TAPEOUT    DD    DSN=NOT . USED , UNIT=TAPE ,
//           LABEL=( , AL ) , DISP=( NEW , KEEP )
//SYSIN      DD    *
ANALYZE
```

**EXAMPLE 13:** 7 files from unlabeled tape 222222 are to be copied to a scratch unlabeled tape. "NUMFILES=7" tells FATAR to copy 7 files (7 tape marks and all data preceding them). Note that "NUMFILES=0" could have been specified or defaulted to cause FATAR to copy all files until two consecutive tape marks were read. The first "KEEP" statement causes block 7 of file 2 to be written to the output tape even though it is known to cause a data check. The second "KEEP" statement causes block 12 of file 3 to be increased in length to 1400 bytes, with the original data block positioned at byte 51 (50 bytes of spaces will be added to the start of the block; if the original block is less than 1350 bytes long, additional spaces will be added to the end). The "VERIFY" parameter will cause the output tape to be analyzed for readability; however, if the output tape requires multiple volumes, the verify will be suppressed (multi-volume NL tapes cannot be verified).

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS , UNIT=TAPE , VOL=SER=222222 ,
//           LABEL=( , NL ) , DISP=OLD
//TAPEOUT    DD    DSN=NOT . USED , UNIT=TAPE ,
//           LABEL=( , NL ) , DISP=( NEW , KEEP )
//SYSIN      DD    *
ANALYZE NUMFILES=7 , VERIFY
KEEP DCK , LF=2 , B=7
KEEP LF=3 , B=12 , L=1400 , P=51
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 14:** File 3 from standard label tape 111111 is to be copied to an unlabeled scratch tape. "LABEL=3" on the TAPEIN DD statement positions FATAR to the proper file; "LABEL=(,NL)" on the TAPEOUT DD statement requests an unlabeled scratch and causes FATAR to copy the TAPEIN data without labels. The "NUMFILES=1" parameter restricts FATAR to copying one file.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=THIRDDS , UNIT=TAPE , VOL=SER=111111 ,
//           LABEL=3 , DISP=OLD
//TAPEOUT    DD    DSN=NOT . USED , UNIT=TAPE ,
//           LABEL=( , NL ) , DISP=(NEW , KEEP)
//SYSIN      DD    *
ANALYZE NUMFILES=1
```

**EXAMPLE 15:** When a tape is to be migrated to a new media, such as a new density, or to IBM 3480 tape cartridges, one way to do it with minimum difficulties is to "IMAGE COPY" the tape. This involves copying all data from the input to the output tape including all labels and the volume serial. Since the output tape will have the same volume serial as the original, in many cases tape management systems will not have to be modified or updated. System catalogs may need to be updated to reflect if the output device is a different device type. "LABEL=(,BLP)" is required on the TAPEOUT DD statement since FATAR is to write all labels. "LABEL=(,BLP)" is optional on TAPEIN. "LABELS=NO" is required so that FATAR will copy TAPEIN labels as data.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=ANYDSN , UNIT=TAPE , VOL=SER=111111 ,
//           LABEL=( , BLP ) , DISP=OLD
//TAPEOUT    DD    DSN=NOT . USED , UNIT=3480 ,
//           LABEL=( , BLP ) , DISP=(NEW , KEEP)
//SYSIN      DD    *
ANALYZE LABELS=NO
```

**EXAMPLE 16:** Standard labeled tapes 333333, 444444, and 555555 comprise a multi-volume, multi-file tape set, containing 17 files at 1600 BPI, and are to be copied to one or more 3480 tape volumes. The "VERIFY" parameter causes the resulting 3480 volumes to be analyzed for readability. The "REPLACE" statement is used to modify the data set name in the header label of the third file from "FILE.THREE" to "THIRD.FILE" ("LF=3H" causes MODIFICATION of the header records of logical file 3); this modified name will be used to open the output file.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPESUMM   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS , UNIT=TAPE16 ,
//           VOL=SER=( 333333 , 444444 , 555555 ) ,
//           DISP=OLD
//TAPEOUT    DD    DSN=NOT . USED , UNIT=3480 ,
//           DISP=(NEW , KEEP)
//SYSIN      DD    *
ANALYZE VERIFY
REPLACE LF=3H , B=1 , P=5 , V=C ' FILE . THREE ' ,
        D=C ' THIRD . FILE ' '
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 17:** A mailing list file on tape has a company name misspelled in many places; we want to copy the file and change the name wherever it appears. The input data set is cataloged; we want to uncatalog it and recatalog the new copy. The "REPLACE" statement causes all occurrences of "INVENTION DATA PROCESSING" to be changed to "INNOVATION DATA PROCESSING" whenever it appears in column 17 of any record. Note the way that the character strings are continued; multiple "C" or "X" strings following a "V=", "S=" or "D=" will simply be concatenated (up to 256 bytes). The "MSGLEVEL=0" parameter suppresses the printing of the data portion of the modified blocks. The "UNCATLG" and "CATLG" in the DISP parameter of TAPEIN and TAPEOUT are possible because we are using the "LABELS=OS" parameter, allowing normal data management label processing to occur.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=COMPANY.MAIL.MASTER,
//           DISP=(OLD,UNCATLG)
//TAPEOUT    DD    DSN=COMPANY.MAIL.MASTER,UNIT=TAPE,
//           DISP=(NEW,CATLG)
//SYSIN      DD    *
ANALYZE LABELS=OS,MSGLEVEL=0
REPLACE LF=1,B=ALL,P=17,V=C'INVENTION DATA',
        C'PROCESSING',D=C'INNOVATION DA',
        C'TA PROC',C'ESSING'
```

**EXAMPLE 18:** Tape master file "MASTER" on labeled tape 11111 causes a run-time data exception when it is read by its processing program. A previous FATAR run has revealed that record 3 of block 27 contains "00735B4A" in hexadecimal, which is an invalid packed decimal number. It should contain "0073534C". The "REPLACE" statement causes this substitution to be made.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=MASTER,UNIT=TAPE,VOL=SER=111111,
//           DISP=OLD
//TAPEOUT    DD    DSN=MASTER,UNIT=TAPE,
//           DISP=(NEW,KEEP)
//SYSIN      DD    *
ANALYZE
REPLACE LF=1,B=27.3,S=X '00735B4A',
        D=X '0073534C'
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 19:** Unlabeled tape 222222 is known to have data checks in a number of blocks. Copy the tape to another unlabeled scratch, retrying all data checks 256 times in each direction to insure that all possible attempts have been made to recover the data. If any block still cannot be read, the "KEEP" parameter will cause it to be written to the output tape anyway, but the "REPLACE DCK" statement will cause the first four bytes of those records to be set to hexadecimal "FFFFFFF" (high values) which will signal the application program to perform special validation. The DCB parameters are provided on the TAPEIN DD statement so that the input records can be deblocked.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=NL INPUT , UNIT=TAPE , VOL=SER=222222 ,
//           LABEL=( , NL ) , DISP=OLD , DCB=( RECFM=FB , LRECL=240 )
//TAPEOUT    DD    DSN=NL OUTPUT , UNIT=TAPE ,
//           LABEL=( , NL ) , DISP=( NEW , KEEP )
//SYSIN      DD    *
              ANALYZE  RETRY=256 , KEEP
              REPLACE  DCK , F=1 , B=a I I , P=1 , D=X  ' FFFFFFFF '
```

**EXAMPLE 20:** Data set "LARGEDS" occupies three tape volumes "AAAAAA", "BBBBBB", and "CCCCCC". Volume "BBBBBB" has developed read data checks, so we want to create a valid copy, dropping the data check blocks. With FATAR, it is not necessary to read the volumes preceding the first data check (in this case "AAAAAA"). The JCL below will produce output volumes which can be used to REPLACE "BBBBBB" and "CCCCCC". However, it is up to the user to update the system catalog to reflect the new volume serials. If you have a tape management system, you must also do whatever manual updating is necessary to cause its records to properly reflect the new volume relationships. Note that this tape management update will be unnecessary if you copy the entire tape set (as in Example 16) or if you copy only the problem tape (as in Example 21).

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=LARGEDS , UNIT=TAPE ,
//           VOL=SER=(BBBBBB , CCCCCC) , DISP=OLD
//TAPEOUT    DD    DSN=LARGEDS , UNIT=TAPE , DISP=( NEW , KEEP )
//SYSIN      DD    *
              ANALYZE
```

**EXAMPLE 21:** This is another way to recover from the data checks in example 20 above. It does not require any catalog or tape management updates since it creates an exact copy of the bad tape, including its volume serial; in other words, the output of this run is a tape with the same volume serial as the original tape, so it should be externally re-labeled and the original tape discarded. Since tape "BBBBBB" is the middle tape of a multi-volume set, we know that it is completely filled with data; the output tape must be as large or larger than the original so that FATAR can WRITE all of the data and labels before the end-of-tape is detected, or the copy will be invalid. FATAR will correct the trailer label block counts if blocks are dropped because of data checks.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=LARGEDS , UNIT=TAPE ,
//           VOL=SER=BBBBBB , DISP=OLD
//TAPEOUT    DD    DSN=LARGEDS , UNIT=TAPE , DISP=( NEW , KEEP ) ,
//           LABEL=( , BLP )
//SYSIN      DD    *
              ANALYZE  NUMFILES=3 , LABELS=NO
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 22:  
RECOVER  
DATA  
FROM AN  
OVERWRITTEN  
TAPE**

A tape with valid data on it was accidentally used for output, but the new file is smaller than the original file; it is desired to recover as much of the original data file as possible. The new file will occupy four physical files on the tape (header labels, data, trailer labels, and extra tape mark), so the "LABEL=(5,BLP)" will position FATAR immediately beyond the extra tape mark, at the beginning of the remaining original data. FATAR will process this as unlabeled data since it is not positioned to label records. The TAPEOUT DD statement specifies a labeled scratch tape with the data set name of the original data set. The first data block is likely to cause a permanent data check if it is partially overwritten. "NF=1" causes FATAR to read only the original data file.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=ORIGINAL , UNIT=TAPE ,
//           VOL=SER=BBBBBB , LABEL=( 5 , BLP ) , DISP=OLD
//TAPEOUT    DD    DSN=ORIGINAL , UNIT=TAPE , DISP=(NEW , KEEP)
//SYSIN      DD    *
ANALYZE NF=1
```

**EXAMPLE 23:  
COPY FILES  
FROM AN  
OVERWRITTEN  
TAPE**

A tape with multiple files on it was accidentally used for output, but the new file is smaller than the original first file; it is desired to copy the original files beyond the first. The new file will occupy four physical files on the tape (header labels, data, trailer labels, and extra tape mark) and the original first file will occupy two more (data file and trailer labels) so the "LABEL=(7,BLP)" will position FATAR to the header labels of the original second file. FATAR will process this as a labeled tape. All data set names from the original tape will be copied to the output scratch tape.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=ORIGINAL , UNIT=TAPE ,
//           VOL=SER=BBBBBB , LABEL=( 7 , BLP ) , DISP=OLD
//TAPEOUT    DD    DSN=NOTUSED , UNIT=TAPE , DISP=(NEW , KEEP)
//SYSIN      DD    *
ANALYZE
```

**EXAMPLE 24:  
COPY AN  
IMPROPERLY  
CLOSED TAPE**

Due to a system crash or power failure, tape 333333 was not properly closed and has no trailing tape mark or labels; it cannot be read with standard access methods. A previous FATAR analyze job shows that block 742 is the last good block of the proper length; a data check occurs immediately after it. The job below will copy the tape up to and including block 742 onto a scratch tape with proper labels.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSUDUMP   DD    SYSOUT=A
//TAPEIN     DD    DSN=A . NAME , UNIT=TAPE ,
//           VOL=SER=333333 , DISP=OLD
//TAPEOUT    DD    DSN=NOT . USED , UNIT=TAPE , DISP=(NEW , KEEP)
//SYSIN      DD    *
ANALYZE ENDAFTER=( LF=1 , B=742 )
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 25:** Multi-file tape 444444 is to be copied to a scratch tape, but files 3 and 5 are not to be copied. The "DROP" statements with "B=ALL" will cause those files, including all of their labels and tape marks to be entirely omitted from the output tape. This means that file 4 from the input tape becomes file 3 on the output, file 6 from the input becomes file 4, file 7 comes file 5, etc. The "VERIFY" parameter causes the output tape to be read back to verify readability, number of files, labels, etc.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSDUMP    DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS,UNIT=TAPE,
//           VOL=SER=444444,DISP=OLD
//TAPEOUT    DD    DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN      DD    *
ANALYZE VERIFY
DROP LF=3,B=ALL
DROP LF=5,B=ALL
```

**EXAMPLE 26:** 7-track tape 777777 is recorded at 800 BPI with even parity, no conversion. It is to be copied to a 9-track scratch. Note that 7-track density and TRTCH must always be specified in the FATAR JCL.

**COPY FROM  
7-TRACK  
INPUT**

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSDUMP    DD    SYSOUT=A
//TAPEIN     DD    DSN=SEVENTR,UNIT=TAPE7,
//           VOL=SER=777777,DISP=OLD,DCB=(DEN=2,TRTCH=E)
//TAPEOUT    DD    DSN=NOT.USED,UNIT=TAPE,DISP=(NEW,KEEP)
//SYSIN      DD    *
ANALYZE
```

**EXAMPLE 27:** Labeled tape "SSSSSS" contains user labels, and we wish to copy it to another labeled scratch tape preserving the labels. Note that the "LABEL=" parameter on the TAPEIN DD statement may specify BLP, SL, SUL, or (if appropriate) AL or AUL; FATAR will recognize the user labels regardless of the JCL label type. Although the output tape specifies "LABEL=(,SL)", FATAR will change it to "SUL" when it detects the user labels on TAPEIN. The output tape may not exceed one volume since FATAR has no way to generate appropriate user labels at volume switch.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//SYSDUMP    DD    SYSOUT=A
//TAPEIN     DD    DSN=ULABDS,UNIT=TAPE,
//           VOL=SER=SSSSSS,DISP=OLD
//TAPEOUT    DD    DSN=NOT.USED,UNIT=TAPE,
//           LABEL=(,SL),DISP=(,KEEP)
//SYSIN      DD    *
ANALYZE
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 28: DROP AN EXTRA TAPE MARK** Because of a malfunctioning tape drive, labeled tape 555555 has an extra tape mark following the trailer labels for file 3. This makes all files past 3 unusable since data management expects an exact sequence of labels and tape marks; FATAR also cannot process the whole tape as labeled, but is capable of processing part of it. Two techniques are shown for recovering from this error:

The first technique causes an exact copy of the tape to be created, but without the extra tape mark. Since the tape mark looks like a file with no data blocks, the "DROP" statement shown will omit that file and therefore drop the tape mark (it is file 10 since the three standard label files preceding it appear to be three files each in "LABELS=NO" mode). The output tape will have the serial number of the original tape and should be externally re-labeled.

```
//FATAR      EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS ,UNIT=TAPE ,
//           VOL=SER=555555 ,DISP=OLD
//TAPEOUT    DD    DSN=NOT .USED ,UNIT=TAPE ,
//           LABEL=( , BLP ) ,DISP=(NEW ,KEEP )
//SYSIN      DD    *
              ANALYZE LABELS=NO
              DROP LF=10 ,B=a l l
```

The second technique involves copying the data up to and beyond the extra tape mark in two separate FATAR steps, but processing in label mode. The first step copies the first three good files. "LABEL=(11, BLP)" in the second step positions the input tape to the header labels of the file beyond the tape mark, and "LABEL=4" on the output tape causes FATAR to start the copy in the right place. The second step then continues to the end of the tape. The output tape will retain its original serial number, and labels will be properly updated if needed.

```
//FATAR1     EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS ,UNIT=TAPE ,VOL=SER=555555 ,
//           LABEL=( 1 , BLP ) ,DISP=(OLD ,PASS )
//TAPEOUT    DD    DSN=NOT .USED ,UNIT=TAPE ,DISP=(NEW ,PASS )
//SYSIN      DD    *
              ANALYZE NUMFILES=3
//FATAR2     EXEC  PGM=FATAR
//SYSPRINT   DD    SYSOUT=A
//TAPEIN     DD    DSN=FIRSTDS ,UNIT=TAPE ,VOL=SER=555555 ,
//           LABEL=( 11 , BLP ) ,DISP=(OLD ,KEEP )
//TAPEOUT    DD    DSN=NOT .USED ,DISP=(NEW ,KEEP ) ,
//           VOL=REF=* .FATAR1 .TAPEOUT ,LABEL=4
//SYSIN      DD    *
              ANALYZE
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 29:** FATAR may be run as a sub-task of FATS, and FATS parameters may be used to call for multiple tape volumes on the same drive. However, each tape is treated as a single volume, not as a multi-volume set. FATAR will be re-initialized for each tape processed. "LABEL=(,BLP)" is used on the TAPE1 DD statement since the tapes will probably have different data set names, and the volume serial "DDDDDD" given on the DD statement is required, but not used. The "VOL=" parameter will cause tapes 111111, 222222, 333333, and 444444 to be copied. A console mount message will be issued for each tape requested. A 3480 scratch output tape will be mounted for each input tape, and all files copied to it. A full FATAR printout will be printed for each tape, and the FATS summary will show the results for all tapes.

```
//FATFTR      EXEC  PGM=FATS
//SYSPRINT    DD    SYSOUT=A
//SYSPRIN1    DD    SYSOUT=A
//SUMMPRT     DD    SYSOUT=A
//SYSUDUMP    DD    SYSOUT=A
//TAPE1       DD    DSN=ANYDSN,UNIT=(TAPE,,DEFER),
//              VOL=SER=DDDDDD,DISP=OLD,LABEL=(,BLP)
//TAPE1OUT    DD    DSN=ANYDSN,UNIT=(3480,,DEFER),
//              DISP=(,KEEP)
//SYSIN       DD    *
              ANALYZE(1) VOL=(111111,222222,333333,444444)
```

**EXAMPLE 30:** When FATAR is run as a sub-task of FATS, FATS may invoke multiple copies of FATAR to perform operations on separate tape drives. Each FATAR execution will produce a separate report on its own "SYSPRINx" DD statement. Multiple volumes may be requested on each drive. The example below will request volumes ABC101, ABC102, and ABC103 on TAPE1, and volumes 000123 and 000125 on TAPE2 (copying the files on them to a scratch tape on TAPE2OUT).

```
//FATFTR      EXEC  PGM=FATS
//SYSPRINT    DD    SYSOUT=A
//SYSPRIN1    DD    SYSOUT=A
//SYSPRIN2    DD    SYSOUT=A
//SUMMPRT     DD    SYSOUT=A
//SYSUDUMP    DD    SYSOUT=A
//TAPE1       DD    DSN=ANYDSN1,UNIT=(TAPE,,DEFER),
//              VOL=SER=DDDDDD,DISP=OLD,LABEL=(,BLP)
//TAPE2       DD    DSN=ANYDSN2,UNIT=(TAPE,,DEFER),
//              VOL=SER=EEEEEE,DISP=OLD,LABEL=(,BLP)
//TAPE2OUT    DD    DSN=NOT.USED,UNIT=TAPE,DISP=(,KEEP)
//SYSIN       DD    *
              ANALYZE(1) VOL=ABC101,VOLINCR=1,MAXVOLN=3
              ANALYZE(2) VOL=(000123,000125),VERIFY
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**EXAMPLE 31:  
COPY AND  
RECATALOG  
MULTIPLE  
FILES TO 3490E  
FORMAT**

The advent of the 3490E tape subsystem presents new challenges for the efficient use of the data center. It would be convenient to reduce the number of tape volumes in the tape library by converting 3420 reels and 3480 cartridges to the new 3490E double capacity compacted format. However, Automatic recataloging of data sets copied by FATAR is normally not possible because:

- 1) The tape labels on tapes read by FATAR contain only the last 17 characters of the data set name, rather than the full 44-character (maximum) data set name known to the system. If the name is longer than 17, FATAR does not know the beginning of that name.
- 2) System catalogs are organized by name, not by volume serial. It is not practical to search catalogs for the tape being copied.
- 3) FATAR does not have access to information maintained by tape management systems, which are organized by volume serial.

If files are copied by normal FATAR techniques and must be cataloged, this cataloging must be done manually using catalog maintenance programs such as IEHPROGM or IDCAMS. Since FATAR does not internally recatalog copied tape data sets, the following special technique must be used to copy and recatalog tapes in one step. The example assumes that the input tape data set names are known and are currently cataloged. The TAPEIN and TAPEOUT DD statements must appear in the order shown so that the uncatlog takes place before the catalog of the file on the new tape. The copy will be successful regardless of the number of volumes occupied by the input file or required for the output file. Note that FATAR will automatically copy the expiration date of each file (even special expirations such as 99000) when opening the output file.

**WARNING: The FDR utility copy program FDRTCOPY should be used to consolidate ABR backup tapes. The FDRTCOPY program has built-in capabilities to control the administration of these tapes.**

**SINGLE FILE  
COPY  
(Use Either  
Technique)**

In this example, the data set names on the input and output tapes are the same, but they don't have to be. If they are different, it can be used to create and catalog a copy of the input file (omitting "UNCATLG" from the input).

```
//COPY3480 EXEC PGM=FATAR
//SYSPRINT DD SYSOUT=A
//TAPEIN DD DSN=BAB.T3480.RECAT.TEST1,
// DISP=(OLD,UNCATLG,KEEP)
//TAPEOUT DD DSN=BAB.T3480.RECAT.TEST1,UNIT=3490,
// DISP=(NEW,CATLG,DELETE)
ANALYZE LABELS=OS
```

In this example, the data set name of TAPEOUT is ignored; the name on the TAPEIN DD statement will be used to open and catalog the output tape. The advantage is that the name needs to be specified only on TAPEIN.

```
//COPY3420 EXEC PGM=FATAR
//SYSPRINT DD SYSOUT=A
//TAPEIN DD DSN=BAB.T3420.RECAT.TEST1,
// DISP=(OLD,UNCATLG,KEEP)
//TAPEOUT DD DSN=DUMMY,UNIT=3490,
// DISP=(NEW,CATLG,DELETE)
ANALYZE NUMFILES=1
```

CONTINUED . . .

## 26.2 CONTINUED . . .

**MULTIPLE FILE COPY** Multi-file tapes (or multi-volume, multi-file tapes) can be copied using a sequence of steps similar to the first example above. As many steps as there are files on the input tape(s) should be created. Note that TAPEOUT in each step refers back to TAPEOUT in the immediately preceding step; failure to do this may result in overlaid files.

```
//COPY3481 EXEC PGM=FATAR
//SYSPRINT DD SYSOUT=*
//TAPEIN DD DSN=BAB.T3480.RECAT.TEST1,
// DISP=(OLD,UNCATLG,KEEP)
//TAPEOUT DD DSN=BAB.T3480.RECAT.TEST1,
// DISP=(NEW,CATLG,DELETE),
// VOL=(,RETAIN),LABEL=1,UNIT=3490
ANALYZE LABELS=OS
//COPY3421 EXEC PGM=FATAR
//SYSPRINT DD SYSOUT=*
//TAPEIN DD DSN=BAB.T3480.RECAT.TEST2,
// DISP=(OLD,UNCATLG,KEEP)
//TAPEOUT DD DSN=BAB.T3480.RECAT.TEST2,
// DISP=(NEW,CATLG,DELETE),
// VOL=(,RETAIN,REF=* .COPY3481.TAPEOUT),
// LABEL=2,UNIT=3490
ANALYZE LABELS=OS
//COPY3422 EXEC PGM=FATAR
//SYSPRINT DD SYSOUT=*
//TAPEIN DD DSN=BAB.T3480.RECAT.TEST3,
// DISP=(OLD,UNCATLG,KEEP)
//TAPEOUT DD DSN=BAB.T3480.RECAT.TEST3,
// DISP=(NEW,CATLG,DELETE),
// VOL=(,RETAIN,REF=* .COPY3421.TAPEOUT),
// LABEL=3,UNIT=3490
ANALYZE LABELS=OS
```

**AUTOMATION OPTIONS** When copying large volumes of tapes, these techniques can be time-consuming, since the data set name of every file to be copied must be known. There are several possible ways to automate the process:

- 1) Tape management systems, if present, record data set names by volume serial and file number. A program could be written to extract tapes to be converted from the tape management system (either via a direct interface or from a listing file generated by that system), and generate the FATAR JCL necessary to copy and recatalog the files involved.
- 2) Although the information in system catalogs is organized by data set name, it is possible to process and sort catalog information (again, either by direct catalog access or by processing a catalog listing) to extract information about tape files to be copied, and generate the FATAR JCL.

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

# FATS<sup>®</sup> & FATAR<sup>™</sup>

USER DOCUMENTATION

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

**30.0 FATS AND FATAR MESSAGES AND CODES****30.1 RETURN CODES**

**RETURN CODES** Both FATS and FATAR use the following completion or return codes:

- 0 – Normal completion. No serious errors
- 4 – A trial version of the product is being executed and the trial period has expired (contact INNOVATION if you wish to continue your testing or install the production version).
- 12 – Serious errors have occurred and the "RETCODE" option has been specified (user abend U0200 or U0888 is issued otherwise).

**30.2 ABEND CODES**

**ABEND CODES** Both FATS and FATAR may issue the following user abend codes (most abends will be accompanied by an explanatory message):

- U0100** (FATAR only) TAPEIN DD statement missing, TAPEIN or TAPEOUT is not assigned to a tape unit, or an open failure on TAPEIN or TAPEOUT
- U0200** (FATAR only) unrecoverable I/O error on TAPEOUT. If the "RETCODE" parameter is specified, a return code of 12 is issued instead of this abend.
- U0401** Open error or I/O error on SYSIN
- U0402** Open error or I/O error on a PRINT data set
- U0502** Error in FATS or FATAR control statements
- U0888** Major errors on tape. For FATS, any condition causing abnormal completion of a FATS function (indicated by "ABNORM" in the FATS summary report). For FATAR, unrecoverable I/O errors on TAPEIN other than data checks, or MAXERR=/MAXTERR= values exceeded. If the "RETCODE" parameter is specified, a return code of 12 is issued instead of this abend.

### 30.3 CONSOLE MESSAGES

**CONSOLE MESSAGES** The following write-to-operator (WTO) and write-to-operator-with-reply (WTOR) messages may be issued by FATS or FATAR. They will be written to route code 3 (tape pool console).

#### **FATSW01 REQUEST TO LABEL TAPE ON UNIT uuu REPLY WITH SERIAL NUMBER**

Reason: The "OPERATOR" parameter has been specified or the "SAVLAB" function found that the tape was not a labeled tape

Action: Respond with a 1-to-6 character volume serial. FATS will left justify it, blank fill it, and label the tape on tape unit "uuu".

#### **FATSW02 REPLY EOJ/KEOJ/Kuuu TO TERMINATE**

Reason: FATS will always issue this WTOR at the start of every FATS run to allow the operator to control and terminate FATS

Action: It will not be necessary to reply to this message unless:  
 1) "MULT" was specified without "MAXVOLN="

- 2) It is desired to terminate FATS before its normal termination, or  
 3) A particular tape must be halted. Valid responses are:
- EOJ – FATS will terminate after all tapes which are currently in progress have completed. Any pending tape mounts must be satisfied.
  - KEOJ – FATS will halt all tapes in progress, cancel any pending mounts, and terminate with a U0888 abend or return code 12.
  - Kuuu – Where "uuu" is the address of a tape drive allocated to FATS. Any mount or tape in progress on that drive will be halted. If appropriate, another tape may be requested on the drive. Other drives allocated to FATS will not be affected. If "uuu" is not a drive in use by FATS, it will be ignored. The FATSW02 message will be reissued.

#### **FATSW03 FATS ANALYSIS OF TAPE ON UNIT=uuu VOL=vvvvvv PERM ERRORS=ppppp TEMP ERRORS=ttttt**

Reason: The "WTO" parameter has been specified. This message will be issued at the end of processing for each tape, giving unit "uuu", volume serial "vvvvvv", and the total number of permanent and temporary errors ("ppppp" and "ttttt"). If a tape has major errors causing it to be halted prematurely, "ppppp" will be "MAJOR".

#### **FATSW04 MOUNT NEXT REEL ON UNIT=uuu VOL=vvvvvv**

Reason: Issued by FATS to request mounting of another tape on tape unit "uuu" under non-MVS operating systems. The volume serial "vvvvvv" may be an explicit serial or "SCRATCH" as appropriate.

Action: Mount the requested tape on unit "uuu".

CONTINUED . . .

## 30.3 CONTINUED . . .

**FATSW05 FATS HAS STOPPED ON DATA CHECK ON UNIT=uuu REPLY CONT, TERM, NOSTOP**

Reason: The "STOP" or "STOPNUM=" parameters have been specified and a permanent data check has occurred. FATS stops to allow visual inspection of the tape at the point of the error.

Action: Inspect the tape on tape unit "uuu", if desired, re-ready the tape, and reply:  
 CONT – To continue certifying/verifying, stopping again on the next data check  
 TERM – To terminate processing of the tape  
 NOSTOP – To continue certifying/verifying without stopping on further data checks

Note that if you remove the tape from the "READY" state to inspect it, you will probably be unable to re-ready it at the same spot and be forced to reply "TERM".

**FATSW06 PLEASE CLEAN UNIT=uuu THEN (RE)MOUNT VOL=vvvvvv**

Reason: The "MAXCLEAN=" or "ERRCLEAN=" parameters were specified. If "MAXCLEAN=n" then "n" tapes have been processed on unit "uuu" since the last cleaning. If "ERRCLEAN=n" then more than "n" errors have occurred on unit "uuu" (the message will say "REMOUNT").

Action: Clean tape unit "uuu" and mount or remount the specified volume "vvvvvv"

**FATSW07 INVALID RESPONSE TO TERMINATE MESSAGE – RE-ENTER**

Reason: The response to message FATSW02 was not "EOJ", "KEOJ", or "Kuuu". FATSW02 will be reissued.

Action: Reply with a valid response for message FATSW02

**FATSW08 VOL=vvvvvv MOUNTED,DSN=d...d,EXPDT=yyddd**  
**FATSW08 UNIT=uuu REQUIRES VOL=xxxxxx-REPLY IGNORE, UNLOAD, TERM, SKIP**  
**FATSW08 UNIT=uuu DSN NOT EXPIRED – REPLY IGNORE, UNLOAD, TERM, SKIP**

Reason: The "VALIDATE=" parameter was specified to validate input and/or output labeled tapes with "LABEL=(,BLP)". Either the wrong volume was mounted or the expiration date on an output volume was not reached. In the first line of the message, "uuu" is the tape unit, "vvvvvv" is the actual volume mounted, "d...d" is the data set name read from the tape, and "yyddd" is the expiration date. One of the next two formats of the message will be issued ("xxxxxx" is the volume expected).

Action: Reply with:  
 IGNORE – To ignore the error and accept the volume mounted.  
 UNLOAD – To unload the volume and request mounting of another tape  
 TERM – To terminate FATS or FATAR  
 SKIP – To terminate this volume and go on to the next (FATS only)

**FATSW11 FATAR FILE ffff OPERATION UNIT=uuu,VOL=vvvvvv,**  
**BLOCKS=bbbbbb,PERM ERRS=ppppp,TEMP ERRS=ttttt**

Reason: The "WTO" parameter has been specified. The message will be issued at the end of processing each input file, giving file number "ffff", unit "uuu", volume serial "vvvvvv", input block count "bbbbbb", and the total number of permanent and temporary errors ("ppppp" and "ttttt") in that file. "Operation" will be "ANALYZED ON" (if TAPEOUT absent), "COPIED FROM" (if TAPEOUT present), or "VERIFIED ON" (if verifying TAPEOUT)

Action: None

CONTINUED . . .

## 30.3 CONTINUED . . .

**FATSW12 FATAR FILE ffff OPERATION UNIT=uuu,VOL=vvvvvv REPLY CONT, TERM, OR NOSTOP**

Reason: The "STOP" parameter has been specified. The message will be issued at the end of processing each input file, giving file number "ffff", unit "uuu", and volume serial "vvvvvv". "Operation" will be "ANALYZED ON" (if TAPEOUT is absent), or "COPIED FROM" (if TAPEOUT present).

Action: Reply with:  
 CONT – To continue with the next input file, stopping again at the end of that file  
 TERM – To terminate processing of TAPEIN  
 NOSTOP – To continue with the next file without stopping at the end of further files

**FATSW13 REPLY IS: TEXT**

Reason: Automatic reply to tape management message.

**FATSW14 INVALID MODIFY COMMAND GIVEN: text....**

Reason: "text" was specified in MODIFY (F) id,text operator command. "id" is the FATS jobname for which the MODIFY command was intended. "text" is not either "QUIT" or "KUUU" to terminate FATS when current tapes are done or to immediately terminate processing for tape unit "uuu".

Action: None. Command is ignored.

**FATSW89 – xxxx – TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRES IN nnn DAYS.****FATSW89 – xxxx – TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRED -----  
PLEASE CONTACT INNOVATION.**

Reason: "xxxx" (FATS or FATAR) trial will expire in "nnn" days or "xxxx" trial has expired. The version of FATS or FATAR that you are running is a trial version and has expired or will expire shortly.

Action: If you have purchased FATS/FATAR and have received the tape, make sure that it was properly installed and the trial version has been deleted. Then rerun the job. If you have not yet purchased FATS/FATAR, call INNOVATION DATA PROCESSING.

**30.4 PRINT MESSAGES**

**PRINT MESSAGES** FATS and FATAR may issue the following messages to their data sets (SYSPRINT or SYSPRINx).

**FATS001 TAPEIN DD STATEMENT MISSING**

Reason: The DD statement for TAPEIN is missing from the job step JCL.

Action: FATAR is terminated with a U0100 abend. Correct the JCL and resubmit the job.

**FATS002 TAPEIN DEVICE IS NOT A TAPE UNIT**

Reason: The DD statement for TAPEIN does not reference a supported tape unit type.

Action: FATAR is terminated with a U0100 abend. Correct the JCL and resubmit the job.

**FATS003 TAPEIN COULD NOT BE OPENED**

Reason: The open of the input tape on TAPEIN failed.

Action: FATAR is terminated with a U0100 abend. Check the printout for operating system messages indicating the cause.

**FATS005 TAPEOUT DEVICE IS NOT A TAPE UNIT**

Reason: The DD statement for TAPEOUT does not reference a supported tape unit type.

Action: FATAR is terminated with a U0100 abend. Correct the JCL and resubmit the job.

**FATS006 TAPEOUT COULD NOT BE OPENED**

Reason: The open of the output tape on TAPEOUT failed.

Action: FATAR is terminated with a U0100 abend. Check the printout for operating system messages indicating the cause.

CONTINUED . . .

## 30.4 CONTINUED . . .

**FATS010 CONTROL CARD ERROR – REASON X**

Reason: A syntax or usage error has been detected in the control statement printed immediately preceding the message. Only one FATS010 message will be printed for each input statement (multiple errors will not be diagnosed). "x" is a letter or number indicating the type of error detected:

- 1 – Open failed for the control statement data set (SYSIN or SYSINx)
- 2 – Control statement is completely blank
- 3 – An invalid operation keyword was found on the control statement
- 4 – More than one "ANALYZE" control statement was found.
- 5 – The control statement required at least one keyword parameter but did not contain one (for all statements except "ANALYZE").
- 6 – A specific keyword parameter was required on the control statement but was not found or a required parentheses was missing or a keyword was misspelled.
- 7 – An "ANALYZE" control statement was required but not found.
- 8 – T= on "TABLE" statement in error. Letter previously used.
- 9 – A numeric value for a keyword parameter exceeds the allowed maximum for that keyword
- A – A value for a keyword parameter was zero or non-numeric
- B – Continuation card required but not found
- C – B= keyword in error. Block range specified but not in ascending order
- D – D=, V=, or S= keyword in error. Length of data is more than 256 bytes or characters.
- E – D=, V=, or S= keyword in error. Data field must contain at least 1 byte or character.
- F – ENDAFTER= keyword error.
- G – Control statement table is too small to contain all information from the control statements present. Increase the table size via the EXEC JCL statement PARM field.
- H – Error in X ' or C' data: no closing apostrophe, invalid HEX digits, or odd number of HEX digits.
- J – Invalid data string. Must start with X ' or C'
- K – Error in T= keyword. On SCAN statement, must be A-Z and defined in a previous table statement (except for P and Z). On table statement, must be A-O or Q-Y and not previously used.
- L – EXITNAME was not found or length of EXITNAME is zero.

Action: FATAR will abend with a U0502 abend code after all control statements have been scanned.

**FATS013 ONE OR MORE ERRORS DETECTED – FATAR TERMINATED**

Reason: Message FATS010 was issued for one or more control statement errors.

Action: FATAR will abend with a U0502 abend code. Correct the errors and re-execute.

CONTINUED . . .

## 30.4 CONTINUED . . .

**FATS014 NO CONTROL CARD DATASET. DEFAULTS ASSUMED**

Reason: No SYSIN DD statement for control statement input was present in the job step JCL (if FATAR was invoked by FATS, no SYSINx DD was present).

Action: All defaults are assumed for all ANALYZE statement parameters. If FATAR was invoked by FATS, any FATAR parameters specified in FATS statements will override these defaults.

**FATS016 ANALYSIS TERMINATED DUE TO MAJOR ERROR**

Reason: An unrecoverable hardware error has occurred on the input tape drive or channel. Preceding messages will give details of the error.

Action: FATAR is terminated with a U0888 abend or return code of 12. Contact INNOVATION if you need assistance in determining the cause.

**FATS017 MAXIMUM ERRORS EXCEEDED – FATAR TERMINATED**

Reason: The number of unrecoverable data checks exceeded the "MAXERR=" parameter, or the number of recoverable data checks exceeded the "MAXTERR=" parameter.

Action: FATAR is terminated with a U0888 abend or return code of 12.

**FATS019 FATAR IS NOT AUTHORIZED TO INTERCEPT 3480 ERROR RECOVERY OR TAPE MANAGEMENT MESSAGES. – FATAR WILL CONTINUE**

Reason: TAPEIN is an IBM 3480 tape cartridge drive, and FATAR must be linkedited and executed as an authorized program to be able to fully intercept and control 3480 tape cartridge errors.

Action: FATAR will continue, but 3480 tape data checks may cause console messages and drive swapping (DDR). You should reply "NO" to any DDR swap messages (console message IGF500 or IGF509). To enable 3480 recovery, re-install FATAR as authorized. In addition, automatic reply to CA-1 tape management messages will be disabled. The legend "\*\*NOT AUTH\*\*" will be placed in the report header lines just before the date. Ordinarily the legend "AUTHORIZED" will appear there.

**FATS020 ANALYSIS TERMINATED AT TAPEMARK SEQUENCE**

Reason: If labels are being processed, two consecutive tape marks were read after a label file. If labels are not processed and "NUMFILES=0" was specified or defaulted, two consecutive tape marks were read.

Action: Processing of the input tape is completed.

**FATS021 ANALYSIS TERMINATED BY EOVLABELS DETECTED IN LABELS=NO MODE**

Reason: If labels are not being processed and "NUMFILES=0" was specified or defaulted, FATAR has detected what appears to be IBM or ANSI EOVLABELS.

Action: Processing of the input tape is completed.

**FATS022 ANALYSIS TERMINATED AFTER SPECIFIED NUMBER OF FILES**

Reason: "NUMFILES=n" was specified, and "n" logical files have been read.

Action: Processing of the input tape is completed.

CONTINUED . . .

## 30.4 CONTINUED . . .

**FATS023 ANALYSIS TERMINATED BY ENDAFTER OPTION**

Reason: The file and block specified by the "ENDAFTER=" parameter was read.

Action: Processing of the input tape is completed.

**FATS024 ANALYSIS TERMINATED BY LOGICAL END OF FILE**

Reason: FATAR attempted to switch to a new input tape volume (due to EOVLABELS read, or end-of-file in "LABELS=OS" mode), but no more volumes were given.

Action: Processing of the input tape is completed.

**FATS025 ANALYSIS TERMINATED BY OPERATOR REQUEST**

Reason: The operator replied "TERM" to the FATSW12 message or FATSW08 message.

Action: Processing of the input tape is halted.

**FATS026 TAPEIN DATA SUCCESSFULLY COPIED TO TAPEOUT VOLUME VVVVVV**

Reason: The TAPEOUT DD statement was present, and reading of TAPEIN data completed successfully. All input data not "DROPPED" has been written to TAPEOUT. The last or only volume of TAPEOUT was "vvvvvv".

Action: None

**FATS027 UNLABELED TAPEOUT REQUIRED MORE THAN ONE VOLUME – VERIFY CANCELLED**

Reason: TAPEOUT is an unlabeled tape ("LABEL=(,NL)" or "(,BLP)"), the "VERIFY" parameter was specified, and TAPEOUT overflowed to more than one volume

Action: FATAR cannot verify multi-volume unlabeled tapes. The VERIFY will not be performed.

**FATS028 ANALYSIS TERMINATED DUE TO INVALID LABEL**

Reason: Labels were being processed on TAPEIN, but a valid IBM standard or ISO/ANSI or user label was not read when one was expected.

Action: The invalid data will be printed, and FATAR will be terminated with a U0888 abend or return code 12

**FATS030 EOT ON TAPEOUT BUT VOLUME SWITCH NOT ALLOWED – FATAR TERMINATED**

Reason: Labels are being processed on TAPEIN, the current file being copied contains user labels, and the "SUPULAB" parameter (suppress user labels) was not specified. User labels may be required with the EOVLABELS (end-of-volume) and header labels written when the file crosses volumes, but FATAR has no way to generate user labels acceptable to the user application.

Action: FATAR is terminated with a U0888 abend or return code 12

**FATS033 PERMANENT I/O ERROR ON TAPEOUT – FATAR TERMINATED**

Reason: FATAR uses standard system error recovery on TAPEOUT. An uncorrectable I/O error occurred (messages will follow to detail the error which occurred).

Action: FATAR will be terminated with a U0200 abend or return code 12. Consult the FATAR messages and operating system messages for details of the error. Errors may be avoided by cleaning the TAPEOUT tape drive, and using an output tape certified by FATS or other means.

CONTINUED . . .

## 30.4 CONTINUED . . .

**FATS034 NOISE BLOCK HAS BEEN READ**

Reason: A data check block too small to be considered a valid data block has been read (will not occur on IBM 3480 tape cartridge drives).

Action: No recovery attempts will be made. The block will be immediately flagged as a permanent data check.

**FATS035 TAPE DRIVE NOT CAPABLE OF REQUIRED DENSITY/MODE**

Reason: The tape drive has reported that it is not capable of reading the input tape at the required density (determined by the "DEN=" and "TRTCH=" JCL parameters for 7-track tape, and from flags on the tape itself for 9-track tape or 3480 cartridges). This error may be caused by an uninitialized tape.

Action: FATAR is terminated with a U0888 abend. Determine the actual density of the tape, and change the TAPEIN DD statement to allocate an appropriate tape drive.

**FATS036 BLOCK GREATER THAN NNNNN BYTES READ – FATAR TERMINATED**

Reason: A tape block longer than "nnnnn" bytes (as specified by the "BLKSIZE=" parameter, DEFAULT 65534) has been read.

Action: The excess data is discarded, and FATAR is terminated with a U0888 abend.

**FATS037 TAPE LABELS IN ISO/ANSI FORMAT – TRANSLATED TO EBCDIC**

Reason: FATAR has detected that the labels on TAPEIN are in ISO/ANSI format ("ANSI labels").

Action: The labels will be translated to EBCDIC for printing or processing. Also, the associated data file is assumed to be in ISCII/ASCII and will be translated to EBCDIC.

**FATS038 DATA FILE IN ISCII/ASCII – TRANSLATED TO EBCDIC**

Reason: The current data file is assumed to be in ISCII or ASCII code. This occurs when "DCB=OPTCD=Q" is specified on the TAPEIN DD card, or when ISO/ANSI labels were detected on TAPEIN by FATAR.

Action: All data blocks in the file will be translated to EBCDIC for printing and processing.

**FATS039 \*\*WARNING\*\* BLOCK LENGTH GREATER THAN BLOCKSIZE OF bbbbb**

Reason: A block was read whose length is greater than the blocksize "bbbb" of the current file (extracted from the header labels of the file, or given on the TAPEIN DD statement if labels are not processed).

Action: None. Unless dropped or modified, the block will be written to TAPEOUT as read.

**FATS040 TAPEIN IS NOT LABELED – LABELS=NO ASSUMED**

Reason: The "LABELS=YES" parameter was specified or defaulted, but FATAR did not find valid IBM or ISO/ANSI labels in the first file read.

Action: FATAR will proceed as if "LABELS=NO" were specified.

CONTINUED . . .

## 30.4 CONTINUED . . .

**FATS042 TAPEIN TRAILER BLOCK COUNT OF bbbbbb IS INCORRECT**

Reason: The EOF1 or EOVI trailer label on a labeled TAPEIN file contained a block count "bbbbbb" which did not match the number of blocks actually read in the file by FATAR. The actual block count is found in the "end of file" message preceding this message.

Action: None

**FATS044 END OF OUTPUT TAPE vvvvvv – NEW VOLUME REQUESTED**

Reason: The end-of-tape reflector (tape indicate) was detected on TAPEOUT volume "vvvvvv".

Action: EOVI labels are written on "vvvvvv" (if labeled), and a new output volume is requested.

**FATS046 \* \* I/O ERROR-> ECB=eeeeeeee ECB rc description**

Reason: This is the first of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the event control block (ECB, 4 bytes) in hexadecimal along with the description associated with the ECB reason code (1st byte of ECB).

Action: Other messages will indicate whether FATAR can continue or must terminate, depending on the nature of the error. Appropriate manufacturer's manuals should be consulted to determine the cause of the error. Contact INNOVATION if you need assistance.

**FATS047 \* \* I/O ERROR-> CSW=wwwwwwww wwwwwwww csw status desc**

Reason: This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the channel-status word. (CSW, 8 bytes) in hexadecimal along with the description associated with the unit and channel status bytes of the CSW (bytes 4 and 5).

Action: See message FATS046

**FATS048 \* \* I/O ERROR-> CCW=cccccccc cccccccw ccw channel cmd**

Reason: This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the channel-command word. (CCW, 8 bytes) in hexadecimal along with the channel command associated with the command code byte of the CCW (byte 0).

Action: See message FATS046

**FATS049 \* \* I/O ERROR-> SENSE ssss sense description**

or

**FATS049 \* \* I/O ERROR-> SENSE ssss EXTENDED SENSE ssss ssssssss ssssssss ssssssss ssssssss ssssssss ssssssss sense description**

Reason: This is one of several messages issued as the result of a non-data check error on TAPEIN or any error on TAPEOUT. For diagnostic purposes it includes the sense bytes (SENSE, 2 bytes) in hexadecimal along with the sense descriptions associated with the first two sense bytes, or, for 3480s, the first sense bytes and the 3480 extended sense bytes (SENSE, total of 64 bytes) along with the ERP (sense byte 3) description.

Action: See message FATS046

**FATS051 RECORD rr BYTE bb LENGTH ll INVALID xxxxx DATA FIELD**

Reason: A SCAN statement has detected data which does not meet the user-specified criteria. "rr" is the record number within the current block ("RECORD rr" will not appear if deblocking is not being done). "bb" is the location of the invalid field within the indicated record or block (relative to 1). "ll" is the length of the field. "xxxx" will be "ZONED", "PACKED" or "TYPE t" (where "t" is a table ID).

Action: None

CONTINUED . . .

## 30.4 CONTINUED . . .

**FATS052 SCAN FIELD NOT IN RECORD – SCAN TERMINATED FOR THIS RECORD**

Reason: A SCAN statement applied to a block or record, but the location to be scanned ("P=" parameter) was beyond the end of the block/record.

Action: SCAN not performed. Check the SCAN parameters.

**FATS053 BLOCK LENGTH ZERO – BLOCK NOT KEPT**

Reason: A KEEP statement applied to an input block, but the length of the data read was zero (probably due to a data check).

Action: The block will not be written to TAPEOUT.

**FATS054 BLOCK LENGTH CHANGED WHILE IN LOGICAL RECORD MODE – DEBLOCKING TERMINATED**

Reason: FATAR was deblocking input blocks into individual records, but a KEEP statement changed the length of a block.

Action: The block will not be deblocked

**FATS056 RECORD TOO SHORT–NO DATA REPLACED**

Reason: A record or block is too short to contain the data specified on a REPLACE statement

Action: REPLACE not performed. Check the REPLACE parameters.

**FATS057 DATA RECOVERED – BLOCK NOT MODIFIED**

Reason: A MODIFICATION/SCAN statement with the "DCK" parameter applied to a data block, but the block did not have a permanent data check.

Action: The MODIFICATION or SCAN will not be performed.

**FATS058 RECORD nnnnn HAS BEEN DROPPED**

Reason: In response to a "DROP" statement with a record number indicated, record "nnnnn" has been deleted from the current block.

Action: The block will be "compressed" to eliminate the deleted record. If the record format is variable, the block length will be updated.

**FATS061 RECFM/LRECL NOT GIVEN – FILE ASSUMED UNBLOCKED**

Reason: The record format (RECFM) and logical record length (LRECL) for the current file are not available from either the TAPEIN DD statement or TAPEIN header labels.

Action: FATAR is unable to deblock the current file into logical records and will treat each block as a single record. If deblocking is desired, specify RECFM and LRECL on TAPEIN and re-execute FATAR.

**FATS062 DEBLOCKING ANSI SPANNED RECORDS NOT SUPPORTED – UNBLOCKED ASSUMED FOR THIS FILE**

Reason: The TAPEIN DD statement or TAPEIN header labels specifies ANSI variable-length spanned records (record format DS or DBS). FATAR does not support deblocking of ANSI spanned records.

Action: Unblocked (record format U) assumed.

CONTINUED . . .

## 30.4 CONTINUED . . .

**FATS063 BLOCK INVALID FOR RECFM/LRECL IN USE-DEBLOCKING TERMINATED**

Reason: Blocks in the current file were being deblocked into logical records, but the current block cannot be deblocked using the record format and logical record length provided (via the TAPEIN DD statement or TAPEIN header labels). For fixed length records, the block is not an exact multiple of the record length. For variable length records, the record descriptor words (RDW) may be invalid or indicate a length greater than the block size.

Action: The current block will be treated as unblocked. If the record format and length given are correct, the block is invalidly formatted; if not, specify the correct RECFM and LRECL on TAPEIN and re-execute FATAR.

**FATS064 FILE NOT ELIGIBLE FOR REBLOCKING-REBLOCKING SUPPRESSED FOR THIS FILE**

Reason: "REBLOCK=" was specified, but the current file cannot be reblocked for one of the following:

- 1) The record format is not fixed or variable
- 2) The record format is variable spanned
- 3) The blocksize is zero
- 4) The logical record length is zero (if the record format is FB or VB)
- 5) The blocksize is greater than the "MAXRBLK=" parameter, if specified.

Action: The file will be copied without reblocking.

**FATS065 RECORD LENGTH TOO LARGE FOR REBLOCKING - REBLOCKING SUPPRESSED FOR THIS FILE**

Reason: "REBLOCK=" was specified, but the logical record length of the current file is larger than the specified reblocking block size.

Action: The file will be copied without reblocking.

**FATS066 THIS FILE WILL BE REBLOCKED TO A BLOCKSIZE OF nnnnn**

Reason: "REBLOCK=" was specified, and the current file can be reblocked. The new block size "nnnnn" will be equal to the reblocking block size for variable length files, or the next lower multiple of the record length for fixed length files

Action: The file will be copied and reblocked.

**FATS067 INPUT DEBLOCKING ERROR WHILE REBLOCKING - FATAR TERMINATED**

Reason: "REBLOCK=" was specified, and the current file was being reblocked, but an error occurred extracting logical records from the current input block (message FATS063 will precede this message).

Action: Since the labels on the output tape reflect the new blocksize, FATAR will be terminated with a U0888 abend or return code of 12. Specify the correct RECFM and LRECL on TAPEIN and re-execute FATAR.

**FATS068 INPUT RECORD TOO LARGE WHILE REBLOCKING - FATAR TERMINATED**

Reason: "REBLOCK=" was specified, and the current file was being reblocked, but an input logical record was larger than the output reblocking blocksize.

Action: Since the labels on the output tape reflect the new blocksize, FATAR will be terminated with a U0888 abend or return code of 12. Specify a larger value for REBLOCK= and re-execute FATAR.

CONTINUED . . .

## 30.4 CONTINUED . . .

**FATS069 RECORD nnnnn SPANNING ERROR – xxxxx OF SEGMENT MISSING**

Reason: The TAPEIN JCL or tape labels specifies variable spanned records for the current file, and FATAR has detected an incomplete spanned record. "xxxxx" will be "START" if the end of a record was found without a corresponding beginning, or "END" if a new record was found before a previous spanned record was complete. "nnnnn" is the record number within the current block where the error was detected (if "xxxxx" is "end", the error is probably actually in the preceding block(s)).

Action: An additional FATAR run may be required to print the appropriate blocks and determine which records are truly in error. Another run using either FATAR "REPLACE" statements to fix the spanning flags, or "DROP" statements to delete the partial records may be used to fix the tape.

**FATS070 CONTROL CARD TABLE SIZE IS n BYTES**

Reason: This message is always printed to document "n", the size of the FATAR control statement storage table. This size defaults to 4096 but can be increased by the "SIZE=" parameter in the FATAR EXEC JCL statement "PARM=".

Action: If this FATAR execution receives message FATS010 reason G, increase the table size.

**FATS071 TAPE BUFFER SIZE IS n BYTES**

Reason: This message is always printed to document "n", the size of the FATAR TAPEIN read buffer. The default size is 65535 bytes but can be reduced by use of the "BLKSIZE=" parameter.

Action: Any blocks on TAPEIN exceeding "n"-1 in length will cause FATAR to terminate (see message FATS036).

**FATS072 TAPEIN DATA WILL BE COPIED TO TAPEOUT**

Reason: The TAPEOUT DD statement is present.

Action: All files and data on TAPEIN will be copied to TAPEOUT unless "DROPPED" by data checks or FATAR control statements.

**FATS074 \*\* WARNING \*\* CONTROL CARD NUMBER N WAS NEVER REFERENCED OR REPLACE-VERIFY NEVER SATISFIED**

Reason: This message is issued at FATAR termination to warn of control statements which have never been acted upon (possibly due to user error). Either the file \$NL or block \$NL on the statement was never encountered on TAPEIN, or, for a REPLACE statement with a "V=" or "S=" parameter, the VERIFY or SCAN never caused any data to be replaced.

Action: Review the control statements to be sure that they were coded correctly. Review the FATAR output to be sure that the desired files and blocks were processed.

**FATS075 SPECIAL EXPIRATION OF yydd1 DETECTED, TAPEOUT EXPDT=yydd2**

Reason: FATAR read labels containing an expiration date "yydd1" recognized as having a special purpose, i.e., 98000 or 99000.

Action: If copying to TAPEOUT, the output file will have expiration "yydd2" (98000 is changed to 00000 but some tape management systems will change this to a default retention).

CONTINUED . . .

**30.4 CONTINUED . . .****FATS100 (INSTRUCTION LINE)**

Reason: If "PARM=I" is present on the FATS EXEC JCL statement, this message will be printed repeatedly on SYSPRINT with the internal FATS instructions.

**FATS102 CONTROL CARD ERROR – REASON=x**

Reason: A syntax or usage error has been detected in the control statement printed immediately preceding the message. Only one FATS102 message will be printed for each input statement (multiple errors on a statement will not be diagnosed). "x" is a letter or number indicating the type of error detected:

- 1 – An option parameter is not followed by a blank or comma.
- 2 – Operation keyword is invalid or the tape number in parentheses is not 1 through 9.
- 3 – Tape number has been previously used on another operation statement.
- 4 – An option parameter is invalid.
- 5 – Error in the MODE= parameter. Value is not 2 hexadecimal digits or is not one of the valid values.
- 6 – Error in the RETRY= parameter. Must be "RETRY=h" or "RETRY=(l,h)". "h" must be 1 to 99. "l" must be 0 to "h".
- 7 – Error in the numeric value of a keyword parameter. Invalid numeric digits.
- 8 – Error in the BPI= parameter, BPI=0.
- 9 – Error in the LABEL= parameter, invalid characters in the label value.
- A – Expected continuation statement not received (previous statement ended in a comma).
- B – No ending apostrophe in OWNERID= parameter.
- C – Error in the THRESHOLD= parameter, greater than 32767.
- D – Label operation statement missing LABEL= or VOL= parameter.
- E – Error in BLKSIZE= parameter, must be less than 65536 and more than twice the tape density (BPI), except for 3480 cartridges.
- F – Error in VOL= parameter. Volume serial longer than 6 characters.
- G – Error in VOL= parameter. More than 682 volume serials specified.
- H – VOL= parameter specified on a default statement.
- I – VOLINCR= parameter specified without VOL= parameter.
- J – Blank control statement.
- K – Error in VOLINCR= parameter. The last volume serial in the VOL= parameter could not be incremented due to insufficient trailing numeric digits.
- L – Error in ENDAFTER= parameter (for FATAR execution).
- M – Invalid numeric value for keyword. Value was zero or greater than 32767.
- N – An option parameter was specified which is invalid for the control statement on which it was found. For instance, MODIFY=YES was specified on a read statement, but it is only valid on a DEFAULT statement.

Action: FATS will abend with a U0502 abend code after all control statements have been scanned.

**FATS104 ONE OR MORE ERRORS DETECTED – FATS TERMINATED**

Reason: Message FATS102 was issued for one or more control statement errors.

Action: FATS will abend with a U0502 abend code. Correct the errors and re-execute.

**FATS105 NO INPUT CARDS – FATS TERMINATED**

Reason: No control statements other than "DEFAULT" and comments were present in the input. At least one operation statement must be present.

Action: FATS will abend with a U0502 abend code.

CONTINUED . . .



## 30.4 CONTINUED . . .

**FATS202 IOB SENSE ECB CSW CCW ADDRESS DCB FAILING CCW**  
**FATS202 (DATA CORRESPONDING TO ABOVE LABELS)**  
**FATS202 FULL SENSE= (UP TO 40 BYTES OF DEVICE SENSE DATA)**

Reason: This message is issued after message FATS201 when major I/O errors occur. At least the first two lines will be printed, the first containing the titles shown and the second the corresponding diagnostic data in hexadecimal. If the complete sense data produced by the device is available, it will be printed in the third line (IBM 3480 tape cartridge drives only).

Action: Major errors may indicate severe problems with the tape being processed or problems with the tape unit being used. Appropriate manufacturer's manuals should be consulted to determine the cause of the error. Contact INNOVATION if you need assistance.

**FATS203** A line of the detail report indicating a tape mark read.

**FATS204** A line of the detail report indicating a data check.

**FATS205** A line of the detail report indicating PRINT threshold met.

**FATS206** A line of the detail report showing the compaction achieved for the current line.

**FATS207** A line of the detail report indicating operator cancelled, maximum errors exceeded, or end-of-tape (tape indicate).

**FATS208** A line of the detail report indicating cleaning action.

**FATS209** A line of the detail report indicating that an IDRC compacted file was sensed.

**FATS210 FATS OPEN/CLOSE SUBTASK ABEND ON DD=TAPEx COMP CODE = ssss uuuuu**

Reason: FATS sub-task which handles OPEN, CLOSE, and FATAR has abended with a system (sss) or user (uuuu) abend.

Action: Use of the tape drive on TAPEx will be terminated. If no other TAPEx DD statements are active, FATS will terminate.

**FATS211 OPERATOR CANCELLED ANALYSIS ON DD=TAPE<sub>n</sub>**

Processing of this tape was cancelled by the operator by a reply of "KEOJ" or "Kuuu" to the FATSW02 console message, or "TERM" or "SKIP" to the FATSW08 console message.

**FATS300** A line of the summary report giving DD name, tape unit address, volume serial, operation performed, termination status, tape length, and data check summary by retry count.

**FATS301 END OF REPORT**

Reason: Printed at the end of all detail and summary reports.

**FATS990 INSTALLED INNOVATION TRIAL WILL EXPIRE ON YY.DDD**

Reason: Product extension has completed successfully.

Required JCL:

```
//EXTEND EXEC PGM=FATEXTND,PARM=XXXX
//STEPLIB DD DISP=SHR,DSN=PRODUCT.LIB
//SYSLIB DD DISP=SHR,DSN=PRODUCT.LIB
//SYSDIAG DD SYSOUT=A
```

PARM will be supplied by INNOVATION.

### 30.5 FATS ACTION MESSAGES

#### **ACTION MESSAGES**

The "ACTION" column of the FATS detail report will contain one of the following messages:

##### **3480 FORMAT FILE SENSED**

The current file is a normal 3480 file.

##### **3480XF FORMAT FILE SENSED**

The current file is an IDRC compacted 3480 file.

##### **3490E FORMAT FILE SENSED**

The current file is a 3490E format file which may or may not be IDRC compacted.

##### **3490E WRAP 2 REVERSAL SENSED**

The current file is a 3490E file which has reached logical EOT and is recorded in the reverse direction.

##### **DATA COMPACTED BY xx%**

The current file is a 3480XF or 3490E file which was IDRC compacted by the indicated amount.

##### **TAPE INDICATE**

The logical end of tape (end-of-tape reflector or 3480 logical end-of-tape) has been detected by a FATS WRITE operation.

##### **PERM DATA CHECK**

An error has been retried until the retry level defining a permanent error ("RETRY=" parameter) has been reached.

##### **PERM DATA CHECK CNTL**

An error was encountered while repositioning the tape for data check retry. No further retries are done. This may indicate serious problems with the tape or drive.

##### **TEMP DATA CHECK**

An error has been retried and was successful before the permanent error retry level was reached.

##### **TAPE MARK**

A tape mark (end of file) was read during a FATS read operation.

##### **LABEL SAVED**

The volume serial was read from the tape and labels were successfully rewritten (the "SAVLAB" function).

##### **LABEL NOT SAVED**

The "SAVLAB" function was attempted, but labels did not exist or could not be read.

##### **LABEL WRITTEN**

Labels with the specified volume serial number have been written to the tape.

##### **LABEL NOT WRITTEN**

Labels could not be written due to I/O errors.

##### **TEMP DATA CHK LABEL**

Temporary I/O errors were encountered when writing labels but the labels were written successfully.

##### **PERM DATA CHK LABEL**

Permanent I/O errors were encountered when writing or saving labels. The labels were not written.

CONTINUED . . .

**30.5 CONTINUED . . .****MAX ERROR EXCEEDED**

The length in inches of a contiguous data check has exceeded the "MAXCERR=" parameter specified. Processing of this tape will be terminated.

**PRINT THRESHOLD MET**

The number of errors printed for this tape exceeds the "THRESHOLD=" parameter specified. No more errors will be printed in the detail report, but processing will continue.

**OPERATOR CANCELLED**

Processing of this tape was cancelled by the operator by a reply of "KEOJ" or "Kuuu" to the FATSW02 console message or "TERM" or "SKIP" to the FATSW08 console message.

**OPEN ERROR OR NO DD**

Errors occurred opening this tape. Processing of all tapes on this tape unit will be halted.

**CLEAN ACTION REQUEST**

The number of errors on this tape exceeds the "ERRCLEAN=" parameter. The operator is requested to clean the tape drive and remount the same tape.

## 40.0 SPECIAL CONSIDERATIONS

This new section will examine special considerations which apply to the use of FATS and FATAR. These may be special cases of using the system in new ways that were not envisioned or descriptions of more complex operations requiring additional knowledge.

### 40.1 VARIABLE SPANNED RECORD ABENDS

S002 abends on variable spanned format tapes (RECFM=VBS or VS) can be very frustrating because of the difficulty in diagnosing and correcting the errors. By far the most common use of VBS format is for SMF/RMF data, since their format is variable with few limitations on the maximum logical record size, so a S002 abend on those tapes may threaten an installation's accounting and performance data.

S002 abends usually indicate an improperly spanned logical record, due to some logical or physical error when creating the tape, or possibly due to blocks dropped when copying with FATAR because of permanent data checks. FATAR is one of the few programs which can detect and correct these spanning errors, but it is a process requiring several passes of the tape and some human analysis.

Variable spanned tapes have the characteristic that logical records may "span" across two or more physical blocks. Because of this, a logical record may be larger than the blocksize of the file. Because VBS format attempts to fill each physical block to capacity, the last logical record in each block, regardless of size, will almost always span to the next block, so almost every block begins and ends with a spanned record.

When a record is broken into pieces for spanning, each piece is referred to as a segment. There are flags in the RDW (record descriptor word, preceding each variable length record) which indicate "first segment", "middle segment", or "last segment" of the total logical record ("middle segment" exists only when a logical record spans three or more blocks).

FATAR automatically performs spanning checks on VS or VBS tapes, detecting missing first or last segments. When FATAR detects a missing segment, it prints the message FATS069 START (or END) OF SEGMENT MISSING. The block and record numbers printed to indicate where the error was detected, but the actual problem is usually in the preceding block. End of segment missing indicates that FATAR found the start of a new logical record before finding the end of the preceding logical record, which started in one of the immediately preceding blocks.

Start of segment missing indicates that FATAR found a middle or end segment not preceded by a start segment, which again should have been in the preceding block. One situation FATAR cannot detect is a missing middle segment where the start and end segments exist; it might even be that those start and end segments do not even belong together because of lost intermediate blocks, but there is no way to detect that. These errors will not usually cause S002 abends, but may cause data errors in the application programs reading them. If you can identify the records causing the error, FATAR may still be used to correct this problem.

### 40.1.1 CORRECTING S002 ABENDS

To correct these problems with FATAR, follow this procedure:

1) Analyze the tape once with FATAR (no TAPEOUT) and note which blocks and records FATAR identifies as having spanning errors (FATS069 messages).

2) Analyze the tape again, printing the complete blocks for at least three blocks before and after each block identified by FATS069, using this input:

```
ANALYZE PRTLEN=32760,DUMP
PRINT LF=F,B=B1-B2
```

substituting the proper file number (F) and a range of block numbers (B1-B2). Include multiple print statements if multiple FATS069 messages occurred.

3) Examine the printout to see where various logical records start and end. FATAR will print each physical block broken down into logical records; it prints the block number and then numbers the logical records (or segments) in the block. You can see the spanning flags in the printout in byte 3 of each segment:

```
00 – complete logical record (not spanned)
01 – start segment
02 – end segment
03 – middle segment
```

4) If you received start of segment missing, then the middle and end segments left must be deleted. Starting at (and including) the block and record number indicated in the FATS069 message, look at each segment until you find the end segment (byte 3 = x '02'). Write down the block number and record number of each segment you examine (including the end segment) since each must be deleted.

5) If you received end of segment missing, then start at the record preceding the one identified in the FATS069 message and look at each record going backwards until you find the start segment (byte 3 = x '01'). Write down the block and record number of each segment examined in order to delete them. Do not include the record from the FATS069 message.

6) Run FATAR again with a TAPEOUT DD statement in order to copy and correct the spanned tape, using this input:

```
ANALYZE VERIFY
DROP LF=F,B=B.R
```

substituting the proper file number (F), block number (B) and record number (R) to be deleted. Include a drop statement for each segment you identified above. The dropped segments should eliminate the spanning errors, and the verify option will cause the output tape to be analyzed to be sure (no FATS069 messages should occur during the verification phase).

# FATS<sup>®</sup> & FATAR<sup>™</sup>

USER DOCUMENTATION



**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

## 90.0 FATS AND FATAR INSTALLATION

### DISTRIBUTION TAPE FORMAT

FATS and FATAR are distributed on either standard label cartridge or a 6250 bpi standard label tape. The tape has a volume serial of FAT43P if it contains a production version of FATS/FATAR, or FAT43T if it contains a trial version. The serial and the production/trial status are clearly marked on the external label of the tape.

The production tape (FAT43P) will contain either FATS alone or FATS and FATAR together depending on your license agreement with INNOVATION DATA PROCESSING.

The trial tape (FAT43T) will always contain both FATS and FATAR plus a utility program FATEXTND. The trial programs will expire (i.e., stop functioning) on the expiration date indicated on the external tape label. If you are a current FATS customer and are testing only FATAR, the FATS version on your trial tape is a production version which will not expire.

The files on the tape are:

File 1: DSN=INSTALL  
DCB=(RECFM=VS,LRECL=3216,BLKSIZE=3220)  
Installation JCL to load and modify.

File 2: DSN=FATFTR  
DCB=(RECFM=VS,LRECL=3216,BLKSIZE=3220)  
IEBCOPY unloaded copy of programs.

File 3: DSN=FATDOC  
DCB=(RECFM=VS,LRECL=3216,BLKSIZE=3220)  
IEBCOPY unloaded copy of user's manual.

File 4: DSN=FATICL  
DCB=(RECFM=VS,LRECL=3216,BLKSIZE=3220)  
IEBCOPY unloaded copy of JCL and examples

File 5: DSN=FATCMD  
DCB=(RECFM=VS,LRECL=3216,BLKSIZE=3220)  
IEBCOPY unloaded copy of dialog CLISTs.

File 6: DSN=FATPNL  
DCB=(RECFM=VS,LRECL=3216,BLKSIZE=3220)  
IEBCOPY unloaded copy of dialog PANELs.

File 7: DSN=FATMES  
DCB=(RECFM=VS,LRECL=3216,BLKSIZE=3220)  
IEBCOPY unloaded copy of dialog MESSAGEs.

File 8: DSN=FATSKE  
DCB=(RECFM=VS,LRECL=3216,BLKSIZE=3220)  
IEBCOPY unloaded copy of dialog SKELETONs.

File 9: DSN=FATTBL  
DCB=(RECFM=VS,LRECL=3216,BLKSIZE=3220)  
IEBCOPY unloaded copy of dialog TABLEs.

CONTINUED . . .

## 90.0 CONTINUED . . .

**INSTALL-  
ATION  
INSTRUCTIONS**

Execute the following JCL to load the INSTALL JCL in file 1 into a JCL or CNTL library of your choice. Two members called INSTALL and INSTMIN will be copied into the target library.

1. Code your standard job statement in place of the dummy job card shown below.
2. Place your unit type in place of the **\*\*TAPE\*\*** shown on the TAPEIN DD statement shown below.
3. Code the volume serial number in place of the **FAT43\*** shown in the TAPEIN DD statement below. Look at your installation tape. It should say either **FAT43T** or **FAT43P** depending on whether you have a trial or production version.
4. Place the name of the data set you've decided will hold the installation jcl in place of the "your.library.here" notice in the LIBRARY DD statement below.

```

JOB CONTROL //INSTALL      JOB      (201,890,7300), 'Place your job card here',
FOR LOADING //                      CLASS=A,                               NOTE 1
INSTALL JCL  //                      MSGCLASS=A,
//                      NOTIFY=YOURID
//COPY          EXEC    PGM=IEBCOPY, PARM=MOD
//TAPEIN        DD      DSN=INSTALL, DISP=OLD, UNIT=**TAPE**,       NOTE 2
//                      VOL=SER=FAT43* (FAT43P OR FAT43T)         NOTE 3
//LIBRARY       DD      DSN=your.library.here,                       NOTE 4
//                      DISP=SHR
//SYSIN        DD      *
//              COPY INDD=((TAPEIN,R)), OUTDD=LIBRARY
//*
```

**NOTE:** The recommended installation procedure is to use the member **INSTALL** to install the Load Library and the FATS/FATAR ISPF Panel Support.

If you want to just install the Load Library then use the member **INSTMIN**.

CONTINUED . . .

90.0 CONTINUED . . .

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

CONTINUED . . .

## 90.0 CONTINUED . . .

**INSTALL-  
ATION JCL  
STREAM**

The JCL stream loaded by the preceding job provides the user with procedures for the installation of INNOVATION's:

...Product load library.  
 ...User documentation file.  
 ...Installation Control library  
 ...ISPF dialog libraries

The JCL stream consists of two in-stream procedures and the necessary steps to execute those procedures to install the FATS/FATAR libraries. The use of procedures allows you to tailor this installation to your site standards.

```

/** ALLOCATE:      PROC TO ALLOCATE A STANDARD LIBRARY
/**
//ALLOCATE        PROC   PROJECT=FATSYS,
//                GROUP=USERID,
//                TYPE=,
//                LIBDA=,
//                LIBVOL=,
//                LIBBLK=6160,
//                LIBPRIM=100,
//                LIBSEC=100,
//                LIBDIR=30,
//                LIBRECL=80,
//                LIBFM=FB
/**
/** ALLOCATE ONE OF FATS/FATAR LIBRARIES
/**
//ALLOCON          EXEC   PGM=IEFBR14
//ALLOCCD          DD     DSN=&PROJECT..&GROUP..&TYPE,
//                UNIT=&LIBDA,
//                VOL=SER=&LIBVOL,
//                DCB=(RECFM=&LIBFM,LRECL=&LIBRECL,BLKSIZE=&LIBBLK),
//                SPACE=(&LIBBLK,(&LIBPRIM,&LIBSEC,&LIBDIR),,ROUND),
//                DISP=(,CATLG,DELETE)
//                PEND
/** COPY:          PROC TO IEBCOPY A LIBRARY FROM TAPE TO DISK
/**
//COPY            PROC   PROJECT=FATSYS,
//                GROUP=USERID,
//                TYPE=,
//                TAPUNIT=,
//                TAPETYP=,
//                SOUT='*',
//                SYSDA=SYSDA,
//                WORK=15,
//                FILENUM='**',
//                VERSION=43T
/**
/** COPY ONE OF FATS/FATAR LIBRARIES
/**
//COPYONE         EXEC   PGM=IEBCOPY
//SYSPRINT        DD     SYSOUT=&SOUT
//SYSUT3          DD     UNIT=&SYSDA,SPACE=(TRK,&WORK)
//SYSUT4          DD     UNIT=&SYSDA,SPACE=(TRK,&WORK)
//LIBIN           DD     DSN=FAT&TAPETYP,
//                LABEL=(&FILENUM,SL,EXPDT=98000),
//                UNIT=(&TAPUNIT,,DEFER),
//                VOL=(,RETAIN,SER=FAT&VERSION),
//                DISP=(OLD,PASS)
//LIBOUT          DD     DSN=&PROJECT..&GROUP..&TYPE,
//                DISP=SHR
//SYSIN           DD     DSN=&&CCARD,DISP=(OLD,PASS)
//                PEND

```

CONTINUED . . .

## 90.0 CONTINUED . . .

The libraries are copied using IEBCOPY with REPLACE. It is probably better to install into a set of new install libraries initially for testing rather than copying over your existing version. If you do copy over your existing version, this installation will replace all like named members. You can first delete those members and compress the data set before installation to reduce space requirements and avoid X37 abends.

1. Code your standard job statement in place of the dummy job card shown below.

```
//INSTALL      JOB   (201,890,7300), 'PLACE YOUR JOB CARD HERE',
//              CLASS=A,
//              MSGCLASS=A,
//              NOTIFY=YOURID
//*
//* note 1 applies to job card
```

2. Both procedures use the symbolic variables PROJECT, GROUP, and TYPE to define data set names for the FATS/FATAR libraries. This allows you to satisfy installation criteria for data set names. The ALLOCATE procedure allows you to allocate new libraries for FATS/FATAR should that be necessary. The COPY procedure allows you to load these libraries or to load the FATS/FATAR members into your libraries.

```
//LOADLIB      EXEC  ALLOCATE,
// PROJECT=FATFTR,TYPE=LOAD,GROUP='**GROUP*',          <=NOTE 2
// LIBDA='**DASD**',LIBVOL='*VOLSER*',                  <=NOTE 3
// LIBBLK=6144,LIBSEC=,                                 <=NOTE 4
// LIBFM=U,LIBRECL=0                                   <=NOTE 5

//COPYDOC      EXEC  COPY,
// PROJECT=FATFTR,TYPE=DOC,GROUP='**GROUP*',          <=NOTE 2
// TAPUNIT='**TAPE**',                                 <=NOTE 6
// SYSDA='**DASD**',                                   <=NOTE 8
// TAPETYP=DOC,FILENUM=3                               <=NOTE 9
```

3. The location of new libraries are controlled by the ALLOCATE procedure symbolic variables LIBDA and LIBVOL.

4. The ALLOCATE procedure uses the symbolic variables LIBBLK, LIBPRIM, LIBSEC, and LIBDIR to control the space allocated for each library. The values provided in the JCL stream can be used as guidelines for space requirements should you desire to use existing libraries.

5. The symbolic variables LIBFM and LIBRECL, along with LIBBLK mentioned in note 4, are used to define the DCB characteristics for new libraries allocated by the ALLOCATE procedure. If you chose to use existing libraries, the RECFM and LRECL should match those shown in the JCL stream for each library.

6. The COPY procedure symbolic variables TAPUNIT and VERSION are used to control the tape allocation and serial number for the IEBCOPY LIBIN DD statement tape allocation. TAPUNIT should be coded with the tape unit type or generic name for the type of installation tape received, either 3420 or 3480. The VERSION should be 43P for a production installation and 43T for a trial installation.

7. The SOUT symbolic variable in the COPY procedure defines the SYSOUT class for installation messages from IEBCOPY.

8. The WORK and SYSDA symbolic variables in the COPY procedure allocate work space for the IEBCOPY steps.

9. Each of the COPY steps defines the tape data set name and file position with the symbolic variables TAPETYP and FILENUM. The FILENUM variable goes into the LABEL parameter. The TAPETYP parameter is used to define the tape data set name.

CONTINUED . . .

## 90.0 CONTINUED . . .

**AUTHORIZING  
FATS/FATAR**

For MVS installations who want to:

- Run FATS against 3480 tape cartridge drives.
- Use the "BLP" control parameters of FATS or FATAR.
- Allow FATS and FATAR to automatically reply to CA1 (UCC-1/TMS) message IECTMS1 and IECTMS2.

The LOAD library created or modified by the JCL above must be defined to MVS as an authorized library so that FATS and FATAR will run as authorized programs. We recommend running authorized (especially for 3480 support), but, if desired, you can copy into a non-authorized library.

**SETTING UP  
THE ISPF  
DIALOG**

Once the FATS/FATAR libraries have been copied to your system you must activate the dialogs. There are several ways to do this, depending on where you copied the dialog components and how you wish to invoke the dialog. If you wish to invoke the FATS/FATAR dialog from an existing selection panel you must use step 1 and step 2. If you wish to invoke the dialog with a separate clist using the ISPF LIBDEF service you must use step 3.

1. Adding the libraries to your logon.

If necessary, make the changes to your logon procedure and/or your logon clist to include the FATS/FATAR dialog libraries. LOGOFF and LOGON again to activate the new allocations.

Proceed to step 2 or 3.

2. Adding the F/F dialog to an existing selection panel.

This step shows you how to add the F/F dialog to an existing selection panel.

EDIT the selection panel you have chosen. Add a selection line for F/F to the ')BODY' section. Use something like the sample line indicated below. This sample is using 'FF' as a two character selection code for the F/F dialog. You can change the FF to any valid ISPF selection code.

```
%FF+ – FATS/FATAR Tape Analysis
```

Add a ZSEL selection line for F/F to the ')PROC' section. Use something like the sample line indicated below. This line becomes part of the ZSEL command.

```
FF,'CMD(FATSFATR OPT("&ZTRAIL")) NEWAPPL(FAT) NOCHECK'
```

The same selection code, 'FF' or your choice, must appear in both the ')BODY' and ')PROC' sections.

The ZSEL command generally looks like:

```
&ZSEL = TRANS(TRUNC(&ZCMD '.'))
code,'action'
"
"
"
FF,'CMD(FATSFATR OPT("&ZTRAIL")) NEWAPPL(FAT) NOCHECK'
' ' ' '
' ' ' '
' * '
' ? '
```

The above example is shown as it might appear with the FATS/FATAR line already in place. SAVE your changes now. At this point you may have to exit ISPF and re-enter it to activate the dialog. Instead, you can enter dialog test, usually option 7 on the primary ISPF panel and return to that panel. This will update the BLDL lists for the dialog libraries to gain access to the changes you have made. Proceed to step 4.

CONTINUED . . .

## 90.0 CONTINUED . . .

3. This step describes how to call the F/F dialog using the ISPF LIBDEF service. For this purpose you can use the provided CLIST, FATLIBDF. Copy FATLIBDF to a SYSPROC library, renaming it according to your own needs.

The PROC statement of the CLIST provides variables which name the libraries required by the FATS/FATAR dialog. This is the PROC statement as delivered:

```

PROC 0      CMD() OPT() APPL(FAT)      +
            PROJ()                    +
            CLIST()                    +
            LLIB()                     +
            PLIB()                     +
            MLIB()                     +
            TLIB()                     +
            SLIB()                     +
            DEBUG

```

EDIT the copied CLIST to provide the library names for the FATS/FATAR components.

If all the libraries have the same high level indexes in their DSNs you only need to change the PROJ()+ line. The CLIST will provide the "standard" low level qualifiers, CLIST, LOAD, PANELS, MESSAGES, TABLES, or SKELETON.

SAVE your changes and proceed to step 4.

4. Once back into ISPF, you may execute the F/F dialog. If it was added to a selection panel go to that panel and select the F/F dialog. If it is to be invoked by a CLIST, type in 'TSO %FATLIBDF' and hit the enter key. If you renamed FATLIBDF use the new name in place of FATLIBDF.

**90.02 PRINTING THE USER MANUAL****PRINTING THE  
USER MANUAL**

The JCL below is supplied on the installation control library with a member name of 'FATPRINT'. Use this JCL to print the manual (if the distribution tape contains trial versions of FATS or FATAR, change the tape serial "FAT43P" to "FAT43T"):

1. Select the appropriate sysout class for your document and change the SYSUT2 DD statement to reflect your choice.
2. 'DSN=FATFTR.\*\*USER\*\*.DOC' must be replaced with the data set name chosen for the documentation file in the INSTALL job.

```
//PRINT EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD DUMMY
//SYSUT2 DD DCB=(RECFM=FA, BLKSIZE=80),
// SYSOUT=* <==USER CHANGE
//SYSUT1 DD DISP=SHR,
// DSN=FATFTR.**USER**.DOC(FATDOC) <==USER CHANGE
```

This JCL can be modified to print multiple copies or to direct the output to a special sysout class or form.

The manual text is formatted using mixed case characters and commonly-available special characters. Special JCL parameters may be required for your impact printer. It is not suitably formatted for laser printers.

No more than 75 characters will be printed on any line, so the manual can be printed on narrow (8.5x11 inch) paper as long as standard 10 characters per inch spacing is used.

**SOURCE  
MODULES**

The installation control library provides one member containing source code:

... OPENEXIT – Sample FATAR OPEN exit which can be used as a model.

## 93.0 SECTION 93 – FATS/FATAR ISPF INTERFACE

FATS/FATAR ISPF panels are available to perform many of the FATS and FATAR functions like labeling, certifying and copying tapes.

The panels will simplify the use of FATS/FATAR for common tasks like certifying or labeling tapes. Extensive editing of the input is done to eliminate control card errors at execution time. More complicated tasks such as dropping blocks or replacing data on a tape will be easier to perform.

**FIGURE 1**  
**FATS/FATAR**  
**MAIN MENU**

Shows the FATS/FATAR Main Menu. You should specify the Profile Command first. The Profile Command will set up the Job Profile and the FATS/FATAR defaults for this USER.ID.

```

----- FATS/FATAR MAIN MENU -----
OPTION ==> P

      A. CERTIFY new tapes
      B. CERTIFY existing labeled tapes whose data sets have expired

      C. LABEL   new tapes

      E. ERASE   old data on tapes for security

      F. VERIFY  whether tapes are still readable           (FATS READ)
      G. VERIFY  whether tapes are still readable           (FATS ANALYZE)
      H. EXAMINE contents and layout (map) of foreign or unknown tapes

      J. COPY    a tape volume      -- (make exact image)
      K. COPY    multiple volumes   -- (make exact image)   (FATS ANALYZE)
      L. COPY    one or more files  -- (make a logical copy)

      M. FATAR   extended functions
      P. PROFILE set up JCL and FATS/FATAR defaults for user

      * Functions G thru M available only if FATAR has been installed

```

**FIGURE 2**  
**FATS/FATAR**  
**JOB PROFILE**

Shows the Panel to enter any Job Control Statements that are required by your installation.

```

----- FATS/FATAR JCL PROFILE -----
COMMAND ==> _

JOB STATEMENT INFORMATION:
==> //CDMW      JOB (ACCOUNT), 'NAME'
==> //*
==> //*
==> //*

LOAD LIBRARIES:

==> FATSYS.RHK.LOAD                                <
==> -----                                     ----- <

SYSOUT CLASS:
==> *

VOLSER DEFAULT:
==> 999999 (default VolSer for Input DD Statement)

Press DOWN for FATS Profile

```

## 93.0 FATS/FATAR ISPF PANELS

**FIGURE 3  
FATS OPTION  
PARAMETERS  
PROFILE**

This is the FATS Option Parameters Profile. Most users will require no changes except for those users who would prefer the lengths that FATS displays in feet to be shown in metric.

```

----- FATS OPTION PARAMETERS PROFILE -----
COMMAND ===> _

MODIFY  -- Respond to operator's MODIFY(F) and STOP(P) commands? : YES (yes/no)
WTOR    -- Issue WTOR message for operator control of processing?: NO (yes/no)
ERRCLEAN -- Clean tape drive after 32767 temporary or permanent errors.
MAXCLEAN -- Clean tape driver after ___ tapes are processed on drive.
MAXERR  -- Stop processing after 200 permanent errors encountered.
MAXCERR -- Stop processing after 200 inches of contiguous permanent errors.
PASS    -- Number of times to write over a tape during certification : 1
RETRY(H) -- Number of retries before error is considered permanent : 10
RETRY(I) -- Number of retries before error is printed in detail report: 1
LINECNT -- Number of lines per page to be printed on FATS reports : 56
NEWPAGE -- Skip to new page when a new volume is processed? : NO (yes/no)
METRIC  -- List tape lengths/error positions in metric units?: NO (yes/no)
RETCODE -- Terminate with return code 12 due to major error? : YES (yes/no)
WTO     -- Report result of each operation on system console?: NO (yes/no)
OWNERID -- Contents of owner ID field in output volume label : _____

          Press DOWN for FATAR Profile      Press UP for JOB Profile

```

**FIGURE 4  
FATAR  
OPTION  
PARAMETERS  
PROFILE**

This is the FATAR Option Parameters Profile. Most users will require no changes except for those users who would prefer the lengths that FATAR displays in feet to be shown in metric.

```

----- FATAR OPTION PARAMETERS PROFILE -----
COMMAND ===> _

MAJERR  -- Stop processing after 1 major errors are encountered.
MAXERR  -- Stop processing after 20 permanent data checks are detected.
MAXTERR -- Stop processing after 100 temporary data checks are detected.
DATAFMT -- Print format of blocks/records: DUMP (char/hex/dump)
LBLPRT  -- Print format of input labels : FORMAT (char/hex/dump/format/none)
PRTLEN  -- Number of bytes/characters of input tape data to be printed: 80
LINECNT -- Number of lines per page to be printed on FATAR reports : 60
METRIC  -- List tape lengths/error positions in metric units?: NO (yes/no)
RETRY   -- Number of retries before error will be considered permanent: 40
RETCODE -- Terminate with return code 12 due to major error? : YES (yes/no)
WTO     -- Report result of each operation on system console?: NO (yes/no)
EOV     -- Check for EOVL labels/update output block counts? : YES (yes/no)
KEEP    -- Copy block even if it has a permanent data check? : NO (yes/no)
EXITNAME -- Name of load module to be executed at open time : _____

          Press UP for FATS Profile

```

## 93.0 FATS/FATAR ISPF PANELS

**FIGURE 5** FATS Certify -- This is Option A from the FATS/FATAR Main Menu.  
**FATS CERTIFY**

```

----- FATS CERTIFY NEW TAPES -----
COMMAND ===> _

TAPE1 DD Dsname=  FATS                                     <
        Unit  =(  TAPE          )
        Label =(  ,BLP,EXPDT=98000          )

VOL    -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                       5: _____  6: _____  7: _____  8: _____
                                       9: _____ 10: _____ 11: _____ 12: _____
                                       13: _____ 14: _____ 15: _____ 16: _____
                                       17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
NOLABEL -- Create no label (NL) tapes: NO (yes/no)
OWNERID  -- Contents of owner ID field in output volume label: _____

```

## 93.0 FATS/FATAR ISPF PANELS

**FIGURE 6** FATS certification of existing tapes whose data sets have expired. This is Option B from the FATS/FATAR Main Menu.  
**FATS CERTIFY EXISTING TAPES**

```

----- FATS CERTIFY EXISTING TAPES -----
COMMAND ===> _

TAPE1 DD Dsname=  FATS                                     <
        Unit  =(  TAPE          )
        Label =(  ,SL          )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                           5: _____  6: _____  7: _____  8: _____
                                           9: _____ 10: _____ 11: _____ 12: _____
                                           13: _____ 14: _____ 15: _____ 16: _____
                                           17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
NOLABEL -- Create no label (NL) tapes: NO (yes/no)
OWNERID  -- Contents of owner ID field in output volume label: _____

```

**FIGURE 7** FATS labeling of new tapes. This is Option C of the FATS/FATAR Main Menu.  
**FATS LABELING OF NEW TAPES**

```

----- FATS LABEL NEW TAPES -----
COMMAND ===> _

TAPE1 DD Dsname=  FATS                                     <
        Unit  =(  TAPE          )
        Label =(  ,BLP,EXPDT=98000          )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                           5: _____  6: _____  7: _____  8: _____
                                           9: _____ 10: _____ 11: _____ 12: _____
                                           13: _____ 14: _____ 15: _____ 16: _____
                                           17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
NOLABEL -- Create no label (NL) tapes: NO (yes/no)
OWNERID  -- Contents of owner ID field in output volume label: _____

```

## 93.0 FATS/FATAR ISPF PANELS

**FIGURE 8**  
**FATS ERASE**  
**FUNCTION**

FATS Erase Function. This is Option E from the FATS/FATAR Main Menu.

**WARNING: This option will erase all data on a tape/cartridge. Use caution when using this option.**

```

----- FATS ERASE OLD DATA -----
COMMAND ==> _

TAPE1 DD Dsname=  FATS                                     <
        Unit  =(  TAPE          )
        Label =(  ,SL          )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                           5: _____  6: _____  7: _____  8: _____
                                           9: _____ 10: _____ 11: _____ 12: _____
                                           13: _____ 14: _____ 15: _____ 16: _____
                                           17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
SAVLAB  -- Preserve volume serial?   : YES (yes/no)
NOLABEL -- Create no label (NL) tapes: NO  (yes/no)
OWNERID  -- Contents of owner ID field in output volume label: _____

```

**FIGURE 9**  
**FATS VERIFY**  
**(READ) OF**  
**EXISTING**  
**DATA ON**  
**TAPE**

FATS verify (read) of existing data on a tape. This is Option F of the FATS/FATAR Main Menu.

```

----- FATS VERIFY (READ) TAPES -----
COMMAND ==> _

TAPE1 DD Dsname=  FATS                                     <
        Unit  =(  TAPE          )
        Label =(  ,BLP,EXPDT=98000          )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                           5: _____  6: _____  7: _____  8: _____
                                           9: _____ 10: _____ 11: _____ 12: _____
                                           13: _____ 14: _____ 15: _____ 16: _____
                                           17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
FILES   -- Number of Physical files to be read: 3

```

## 93.0 FATS/FATAR ISPF PANELS

**FIGURE 10** FATS verify (analyze) of tapes. This is Option G of the FATS/FATAR Main Menu.  
**FATS VERIFY**  
**(ANALYZE) OF**  
**TAPES**

```

----- FATS VERIFY (ANALYZE) TAPES -----
COMMAND ===> _

TAPE1 DD Dsname=  FATS                                     <
Unit  =(  TAPE      )
Label =(  ,BLP,EXPDT=98000                                )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                           5: _____  6: _____  7: _____  8: _____
                                           9: _____ 10: _____ 11: _____ 12: _____
                                          13: _____ 14: _____ 15: _____ 16: _____
                                          17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN  -- Maximum number of volumes : _____
NUMFILES -- Number of Logical files to be read: 0 (zero=all files)
PRINTOPT (L)-- Read and Print tape Labels ONLY: NO (yes/no)
          (B)-- Number of blocks to print for each file processed: _____

```

**FIGURE 11** Lets you examine (MAP) the contents of any tape. This is Option H of the FATS/FATAR Main Menu.  
**EXAMINE**  
**CONTENTS**  
**(MAP) OF**  
**FOREIGN OR**  
**UNKNOWN**  
**TAPES**

```

----- FATAR EXAMINE (MAP) TAPES -----
-----
COMMAND ===> _

TAPEIN DD Dsname=  FATAR                                     <
          Unit  =(  TAPE      )
          Label =(  ,BLP,EXPDT=98000                                )
          VolSer =(  DUMMY                                           )

NUMFILES -- Number of Logical files to be read: 0 (zero=all files)
BLP      -- By-pass tape label processing      : YES (yes/no)
PRINTOPT (L)-- Read and Print tape Labels ONLY: NO (yes/no)
          (N) -- Number of blocks to print for each file processed: _____

```

## 93.0 FATS/FATAR ISPF PANELS

**FIGURE 12** FATAR image copy of a tape. This is Option J of the FATS/FATAR Main Menu.  
**MAKE AN EXACT IMAGE OF A TAPE VOLUME**

```

----- FATAR IMAGE COPY -----
COMMAND ===> _

TAPEIN DD Dsname=  FATARIN.CDMW          <
        Unit  =(  TA80   )
        VolSer=(  999999   )

TAPEOUT DD Dsname=  FATAROUT.CDMW        <
        Unit  =(  TA80   )
        VolSer=(  _____ )

NUMFILES -- Number of Logical files to be read : 0      (zero=all files)
VALIDATE -- Verify Volume Serial/Expiration Date: ALL  (no/input/output/all)
VERIFY   -- Verify readability of Output tape   : YES   (yes/no)

```

**FIGURE 12A** FATAR multiple volumes image copy. This is Option K of the FATS/FATAR Main Menu.  
**IMAGE COPY OF MULTIPLE VOLUME TAPE AGGREGATE**

```

----- FATS MULTIPLE VOLUMES IMAGE COPY -----
COMMAND ===> _

TAPEIN DD Dsname=  FATSIN.CDM5          <
        Unit  =(  TA80   )

TAPEOUT DD Dsname=  FATSOUT.CDM5        <
        Unit  =(  TA80   )

VOL      -- Volume serial number(s)  1: _____  2: _____  3: _____  4: _____
                                       5: _____  6: _____  7: _____  8: _____
                                       9: _____ 10: _____ 11: _____ 12: _____
                                       13: _____ 14: _____ 15: _____ 16: _____
                                       17: _____ 18: _____ 19: _____ 20: _____

MAXVOLN -- Maximum number of volumes : _____
VALIDATE -- Verify Volume Serial/Expiration Date: _____ (no/input/output/all)
VERIFY   -- Verify readability of Output tape   : YES   (yes/no)

```

**FIGURE 13** Make a logical copy of one or more files with FATAR. This is Option L of the FATS/FATAR Main Menu.  
**MAKE A LOGICAL COPY OF ONE OR MORE FILES**

```

----- FATAR LOGICAL COPY -----
COMMAND ===> _

TAPEIN DD Dsname =  FATAR          <
Unit   =(  TAPE   )
Label  =(  ,SL,EXPDT=98000   )
VolSer =(  DUM-
MY
_____ )
DCB    =(  _____ )

TAPEOUT DD Dsname =  FATAR        <
Unit   =(  TAPE   )
Label  =(  ,SL,EXPDT=98000   )
VolSer =(  _____ )
DCB    =(  _____ )

NUMFILES -- Number of Logical files to be read: 0      (zero=all files)
BLP      -- By-pass label processing on Input : NO     (yes/no)
OUTBLP   -- By-pass label processing on Output: NO     (yes/no)
VERIFY   -- Verify readability of Output tape : YES   (yes/no)
LABELS   -- Label checking to be done by FATAR: YES   (yes/no/os/cat)

```

## 93.0 FATS/FATAR ISPF PANELS

FIGURE 14 FATAR Extended Functions. This is Option M on the FATS/FATAR Main Menu.

FATAR  
EXTENDED  
FACILITIES

```

----- FATAR EXTENDED FUNCTIONS -----
OPTION ==> _

1. -- to ANALYZE data on tape
2. -- to DROP (Delete) selected blocks or records
3. -- to KEEP selected blocks or change block length
4. -- to PRINT selected blocks
5. -- to REPLACE (Search for) data in selected blocks or records
6. -- to SCAN data fields in selected blocks or records

```

FIGURE 15 FATAR TAPEIN and TAPEOUT (if required) for the ANALYZE, DROP, KEEP, REPLACE and SCAN operations.

SAMPLE  
TAPEIN AND  
TAPEOUT

```

----- FATAR INPUT FILE FUNCTIONS -----
COMMAND ==> _

TAPEIN DD Dsname=  FATARIN.CDMW          <
        Unit  =(  TA80          )
        Label =(  ,BLP,EXPDT=98000      )
        VolSer=(  _____          )
999999          )
        DCB   =(  _____          )

COPYFILE -- Do you wish to create an Output Tape?: ____ (yes/no)

NUMFILES -- Number of Logical files to be read : 0      (zero=all files)
BLP      -- By-pass label processing on Input   : NO     (yes/no)
LABELS   -- Label checking to be done by FATAR? : YES    (yes/no/os/cat)
VALIDATE -- Verify Volume Serial/Expiration Date: INPUT (no/input/output/all)
STOP     -- Provide operator control at Eof?    : NO     (yes/no)
ENDAFTER(LF)- Stop processing after logical file: _____
          (B) - Stop processing after block number: _____

```

```

----- FATAR OUTPUT FILE FUNCTIONS -----
COMMAND ==> _

TAPEOUT DD Dsname=  FATAROUT.CDMW        <
        Unit  =(  TA80          )
        Label =(  ,BLP,EXPDT=98000      )
        VolSer=(  _____          )
          )
        DCB   =(  _____          )

OUTBLP   -- By-pass label processing on Output   : YES    (yes/no)
MAXRBLK  -- Largest current blksize for reblocking: _____
REBLOCK  -- Target blocksize for data reblocking : _____
COMBFILES -- Combine all input into one output file: NO    (yes/no)
VALIDATE -- Verify Volume Serial/Expiration Date : OUTPUT (no/input/output/all)
VERIFY   -- Verify readability of Output tape   : YES    (yes/no)

```

## 93.0 FATS/FATAR ISPF PANELS

**FIGURE 16** This is Option 2 on the FATAR Extended Functions Panel.  
**FATAR DROP OPERATION**

```

----- FATAR DROP OPERATION #2 -----
COMMAND ===> _

Drop Data Check Block ONLY: N (Y/N)

Logical File..... ALL: N (Y/N)   -OR-   File: _____ Hdr/Trl: __ (H/T)
Blocks/Records..... ALL: N (Y/N)   -OR-   Blk1: _____ . Rcd1: _____
                                           Blk2: _____ . Rcd2: _____

```

**FIGURE 17** This is Option 3 on the FATAR Extended Functions Panel.  
**FATAR KEEP OPERATION**

```

----- FATAR KEEP OPERATION #1 -----
COMMAND ===> _

Keep Data Check Block ONLY: N (Y/N)

Logical File..... ALL: N (Y/N)   -OR-   File: _____ Hdr/Trl: __ (H/T)
Blocks..... ALL: N (Y/N)   -OR-   Blk1: _____ - Blk2: _____

New Block Length..... : _____

Starting Location..... : _____

Reposition Data..... : L (L -- Justify Left)
                       (R -- Justify Right)

```

**FIGURE 18** This is Option 4 on the FATAR Extended Function Panel.  
**FATAR PRINT OPERATION**

```

----- FATAR PRINT OPERATION #1 -----
COMMAND ===> _

Print ALL Blocks: N (Y/N)
----- OR -----
Logical File..... ALL: N (Y/N)   -OR-   File: _____ Hdr/Trl: __ (H/T)
Blocks..... ALL: N (Y/N)   -OR-   Blk1: _____ -- Blk2: _____
Data Print Length..... : _____
Data Print Format..... : _____ (Char/Hex/Dump)

```

93.0 FATS/FATAR ISPF PANELS

FIGURE 19 This is Option 5 on the FATAR Extended Function Panel.

**FATAR  
REPLACE  
OPERATION**

```

----- FATAR REPLACE OPERATION #1 (Panel 1 of 2) -----
COMMAND ===> _

Replace Data Check Block ONLY: N (Y/N)
Logical File..... ALL: _ (Y/N)   -OR-   File: _____ Hdr/Trl: _ (H/T)
Blocks/Record..... ALL: _ (Y/N)   -OR-   Blk1: _____ . Rcd1: _____
                                           Blk2: _____ . Rcd2: _____

Starting Location..... : _____

Scan until End of Rcd/Blk... : N (Y/N)

Scan Option..... : _ (S -- for Search)
                  (V -- for Verify)

Replace Data Option..... : N (Y/N)
    
```

**FIGURE 19  
(CONT.)  
FATAR  
REPLACE  
OPERATION**

```

----- FATAR REPLACE OPERATION #1 (Panel 2 of 2) -----
COMMAND ===> _

SEARCH STRING -      (* Note: Character Type: C for Ebcdic, X for Hexadecimal )
Type: _ >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
REPLACEMENT DATA --
Type:
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    - >                                                    <
    >>>> Enclose your character string(s) in Quotes (ex: 'xx...xx') <<<<<
    
```

93.0 FATS/FATAR ISPF PANELS

**FIGURE 20** This is Option G on the FATAR Extended Function Panel.  
**FATAR SCAN OPERATION**

```

----- FATAR SCAN OPERATION #1 -----
COMMAND ===> _

Scan Data Check Block ONLY: N (Y/N)
  Logical File.....ALL : _ (Y/N) -OR- File: _____ Hdr/Trl: _ (H/T)

Blocks/Record.....ALL : _ (Y/N) -OR- Blk1: _____ Rcd1: _____
                                         Blk2: _____ Rcd2: _____

  Length of Field..... : _____
  Starting Location..... : _____
  Table Name..... : _ (P -- scan for Packed decimals)
                   (Z -- scan for Zoned decimals)
                   (t -- scan according to Table <t>:
                       where <t> = 'A' to 'Y' <excl. 'P'>)

```

**FIGURE 20**  
**FATAR TABLE -**  
**A- DEFINITION**

```

----- FATAR TABLE -A- DEFINITION -----
COMMAND ===> _

SCAN OPERATION: #1

Report/Print only if Character FOUND: N (Y/N)

Table String-- (* Note: Character Type: C for Ebcdic, X for Hexadecimal )

Type: _ > <
      - > <
      - > <
      - > <
      - > <
      - > <
      - > <
      - > <
      - > <
      - > <
      >>>> Enclose your character string(s) in Quotes (ex: 'xx...xx') <<<<<

```

**THIS PAGE WAS INTENTIONALLY LEFT BLANK**

