

FATS/FATAR & FATSCOPY

C O N C E P T S & F A C I L I T I E S G U I D E

FATS/FATAR & FATSCOPY

t a b l e o f c o n t e n t s

Introduction	2
PART ONE. What is FATS/FATAR/FATSCOPY?.....	4
Introduction.....	4
What is FATS/FATAR/FATSCOPY?.....	4
FATS—The “Media Tool”	5
FATAR—The “Data Tool”	5
FATSCOPY—The “Conversion and Stacking Tool”	6
PART TWO. FATS.....	7
Introduction.....	7
Tape Labeling	7
Tape Certification.....	8
Tape Verification	10
Tape Erasure	11
Summary	12
PART THREE. FATAR & FATSCOPY.....	13
Introduction.....	13
Mapping Tapes	13
Copying Tapes—Basic Features	16
Copying Tapes—FATSCOPY	18
Error Recovery	21
Summary	23
PART FOUR. The ISPF Panels.....	24
Introduction.....	24
The Main Menu	24
Sample FATS Panels.....	25
Sample FATAR Panels	26
Summary	27

INTRODUCTION

CONCEPTS & FACILITIES GUIDES

For more than 30 years, Innovation Data Processing has been producing high-quality Storage Management Software. Over the years, its products have evolved into today's ultra high-speed, safe, reliable storage management solutions for OS/390, z/OS, LAN and Open Systems Data.

It all started with the **FDR Storage Management Family**, of which over 5000 licenses have now been sold worldwide. The FDR Family is the complete Storage Management System for OS/390 and z/OS.

FDR has become the industry standard for fast, reliable backups of MVS OS/390 data.

ABR adds a layer of automation to the standard functions of FDR, providing advanced backup facilities like *Incremental Backup*, *Application Backup* and *Archiving*.

COMPAKTOR and **FDRREORG** further enhance the suite by adding intelligent and powerful reorganization processes, for whole DASD volumes and for Sequential, PDS and VSAM datasets.

FDREPORT provides extensive customized DASD Management Reporting to suit many needs and purposes.

FDRCLONE is an extension to ABR, providing the ability to "clone" volumes and/or datasets on a test or disaster recovery system. It includes **FDRDRP**, a utility that can reduce ABR full-volume recovery time by up to 80%.

FDRINSTANT enables FDR/ABR to take *non-disruptive backups* of offline volumes, created by the latest DASD Subsystem features like StorageTek/IBM SnapShot Copy, EMC² TimeFinder/BCV, HDS ShadowImage and IBM FlashCopy.

FDRPAS (FDR Plug and Swap) allows for the non-disruptive movement of OS/390 disk volumes from one disk device to another. When new disk subsystems are installed, active online disk volumes can be swapped to drives in the new subsystem without disrupting normal operations or requiring a re-IPL. This allows a 24 x 7 installation, with no window for major re-configurations and hardware changes, to install and activate new hardware.

THE FDR/UPSTREAM Family of Products builds on the strengths of the FDR Storage Management Family providing a fast, safe and reliable solution to backing up Open Systems data from file servers and workstations, across a network connection to disk or tape on the OS/390 host. If the Open Systems data is resident on an EMC² Symmetrix with Enterprise Storage Platform (ESP), **FDRSOS** and **FDR/UPSTREAM/SOS** products provide additional performance enhancements to the backup and restore process by utilizing high-speed mainframe channels.

IAM is Innovation's alternative to VSAM KSDS, ESDS and (as a cost option) AIX files. It eliminates VSAM performance bottlenecks and reduces VSAM file sizes by more than 50%.

FATS/FATAR and **FATSCOPY** are a set of multi-purpose tape subsystem Media Integrity tools that allow for online tape certification, verification and erasure, as well as the ability to analyze and copy tapes.

INTRODUCTION

CONCEPTS & FACILITIES GUIDES

Each of the Innovation products are described in a range of Concepts & Facilities Guides that have been created by the Innovation UK office, but which are available *free of charge* from your local office (see back cover for details).

In this particular guide, we look at **FATS/FATAR** and **FATSCOPY**. FATS (Fast Analysis of Tape Surfaces), FATAR (Fast Analysis of Tape and Recovery) and FATSCOPY are magnetic tape utility programs that provide a number of useful functions.

PART ONE of this guide is an overview of the three utilities, explaining how FATS can be used to label, erase, certify and verify magnetic tape volumes, while FATAR can analyze tape contents, rescue data from damaged tapes and copy data from older to newer media. FATSCOPY is a cost-option feature of FATS/FATAR, which uses the copying facilities of FATAR to automate stacking of datasets on tape to improve the utilization of high-capacity tape volumes and simplify the migration to new media types.

PARTS TWO and **THREE** then provide more detailed information on how all of these functions can be run, including some example JCL and job output. **PART FOUR** then illustrates the ISPF Panels that are provided with product.

Any comments or suggestions regarding this guide can be directed to:
support@fdinnovation.com

PART ONE

WHAT IS FATS/FATAR/FATSCOPY?

Introduction

It wasn't so long ago that people were predicting the decline in the use of magnetic tape as a storage medium. However, the development of IBM Magstars, StorageTek 9840/T9940s and other similar high-capacity media has proved that these predictions were incorrect. It would be fair to say that the majority of z/OS and OS/390 sites still rely heavily on magnetic tape for one or more of the following:

- Day-to-day backups (dataset/application backups or full-volume backups)
- Disaster Recovery backups
- Migration/archive storage
- Data receipt and/or transfer

Magnetic tape is still an important tier in the z/OS and OS/390 storage hierarchy. And the commercial value of the data being stored on magnetic tape is still as high as it has ever been—from daily backups of company-critical data, to the long-term storage of sensitive financial data that has to be retained for legal reasons for up to seven years. It must be assumed that all of this data is being stored for good reason and that any loss of access to the data could have quite serious implications.

The explosion in tape capacities over the past few years also cannot be ignored. At their introduction in the mid-1980's, 3480 cartridges held 200MB of data, but by the year 2000, cartridges of exactly the same physical size held 80GB (80,000MB) or more. The exposure to data loss due to magnetic tape media failure is, therefore, now *significantly greater* than at any time in the past—a failure on an IBM Magstar or StorageTek 9840/T9940 can involve the loss of many gigabytes worth of data.

Today, more than ever, a product like FATS/FATAR and FATSCOPY are a 'must-have' for all z/OS and OS/390 sites that use magnetic tape.

What is FATS/FATAR/FATSCOPY?

FATS/FATAR and FATSCOPY are a special set of magnetic tape utility programs that can be executed under z/OS and OS/390, either as a batch job or through a supplied set of ISPF panels. They utilize the actual tape drives where the magnetic tapes will be used, rather than relying on expensive stand-alone hardware devices, which may require their own separate maintenance agreements.

All current forms of magnetic tape hardware are supported, from reel-to-reel 3420's and standard 3480/3490 cartridges, to IBM 3590 Magstar (including 3590-H, triple density), and StorageTek 9840/T9940s.

Virtual tape volumes in an IBM 3494 VTS, or created by the EMC CopyCross product, are also supported. However, although the reading, labeling and verification features of FATS/FATAR are still valuable on these virtual tapes, there is no point certifying or erasing them since the data is buffered on an internal disk.

As you will see from this guide, FATS/FATAR and FATSCOPY are *not* Tape Management Systems and they are not designed to replace products like CA-1, CA-TLMS or RMM. Instead, they provide valuable utility functions that help to monitor and maintain the physical quality of the magnetic tape media and the data that is contained on it.

A brief summary of the functions available within FATS, FATAR and FATSCOPY now follows. Then in PARTS TWO and THREE, we go into more detail on how the various functions are executed.

PART ONE

WHAT IS FATS/FATAR/FATSCOPY?

FATS—The “Media Tool”

FATS (*Fast Analysis of Tape Surfaces*) is a media tool designed to provide various utility functions for magnetic tape. The primary functions of FATS are:

- **Certifying** the usability of tapes, both new and old, by writing a pattern on the full length of the tape. This avoids “write data” errors at application run-time by identifying potentially defective tapes. This is equivalent to the function of hardware tape certifiers, but utilizes existing tape drives.
- **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 (see below).

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

FATS also has some valuable secondary functions, such as:

- **Labeling** magnetic tape media with specified volume serials. This can be done by itself or in combination with the certification process mentioned above.
- **Erasing** tapes to ensure that confidential information is not left on a tape (*i.e.* 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.

PART TWO shows examples of the FATS verify, certify, label and erase functions.

FATAR—The “Data Tool”

Where FATS is a ‘media tool’, designed to ensure the continued physical quality of the magnetic tape library, FATAR (*Fast Analysis of Tape And Recovery*) can best be described as a ‘data tool’.

It is a multi-purpose magnetic tape utility that can read any magnetic tape that is usable by z/OS or OS/390, processing multiple files and multiple volumes in one execution. While reading a tape, FATAR offers the following functions:

- It can **MAP** the tape, providing either a summary or detailed report on its contents, including the characteristics of each data file. This feature is extremely useful for investigating “unknown” tapes to discover their label type, file count, DCB characteristics, etc.

During the mapping process, FATAR can also:

- Verify that each file on the tape is properly formatted. It does this by checking every block against the DCB information.
- Detect (and correct—see below) invalid variable spanned records.
- Verify that certain data fields contain valid data.

PART ONE

WHAT IS FATS/FATAR/FATSCOPY?

- FATAR can **COPY** tapes.
 - This may be an exact 'bit-for-bit' image copy of the input tape, copying the files and the volser to the output tape.
 - It can be a 'partial' copy of a tape, copying only selected files.
 - During the copy process, the blocksize and/or label type may be altered (e.g. AL-to-SL).
- FATAR also includes **ERROR RECOVERY** facilities, allowing it to read (and copy) tapes that would otherwise cause problems for normal application programs. For example:
 - It can create a good copy of a tape that has been improperly closed and which may be missing EOF/EOT markers.
 - It can recover data from a tape that has been partially overwritten (in error) by a shorter file. The remaining data can be copied to a new file on a new tape.
 - It can recover (or read past) data checks on a tape, creating a copy of the input tape(s) with the bad data either corrected or dropped.

As highlighted in the Introduction to this section, data that is stored on magnetic tape must be written correctly and still be readable when needed. Many man-hours and a considerable amount of computer time can be wasted trying to recover from physical (hardware) or logical (data) errors on tape. With the above set of tools, FATAR can greatly reduce this recovery time. PART THREE shows some example FATAR jobs, illustrating just how easy it is to use.

FATSCOPY—The “Conversion and Stacking Tool”

FATSCOPY is a cost-option feature of FATS/FATAR, providing the following facilities:

- FATSCOPY can **CONVERT** tape datasets from one magnetic media type to another.
 - It can automate the conversion of tape datasets from one tape media type to another. For example, files currently stored on standard 3480s can be automatically selected (via a range of selection criterion) and copied over to IBM 3590 Magstar and StorageTek 9840s.
- FATSCOPY can automate the **STACKING** of tape datasets on high-capacity volumes.
 - It can periodically collect tape datasets from multiple tapes and stack them onto fewer tapes, releasing the inefficient input volumes as scratch tapes for reuse. The stacked datasets will be re-cataloged, making the stacking process completely transparent to the application.

PART THREE shows some example FATSCOPY jobs.

PART TWO

FATS

Introduction

The FATS functions outlined in PART ONE can be executed by running a batch job under z/OS or OS/390. This section of the guide shows some examples of those batch jobs, illustrating how easy FATS is to use. PART FOUR shows how the same functions can be requested through a supplied set of ISPF panels.

Tape Labeling

FATS labels a tape by writing an IBM standard (SL) or ANSI V3 (AL) label set. This can be done as a stand-alone process, or during the certification or erasure of a tape (see later sections). FATS can also create unlabeled tapes (NL), writing only a tapemark at the beginning of the tape. FATS is ideal for initializing a sequence of new tapes, or re-labeling old tapes with new serials. Here is a sample job to show how FATS can be used to label a batch of new tapes.

Sample FATS LABEL job:

```
//LABEL EXEC PGM=FATS, REGION=OM
//SYSPRINT DD SYSOUT=*
//SYSPRINT1 DD SYSOUT=*
//SUMMPRT DD SYSOUT=*
//TAPE1 DD DSN=FATS, UNIT=(TAPE, , DEFER),
// LABEL=(, BLP), DISP=(, KEEP)
//SYSPRINT DD *
```

LABEL(1) VOL=(TST900) , VOLINCR=1, MAXVOLN=100

100 new tapes will be labeled starting at TST900 and incrementing to TST999 (as determined by the MAXVOLN=100 and VOLINCR=1 parameters). Some other points to note are:

- The serial number can be specified in the Control Statement (as shown in the example) or in JCL on the TAPEX DD statement, or via the Operator's console in reply to a message. A tape that already has a label can either retain that label or be assigned with a new one.
- The labels that FATS writes are fully compatible with IEHINIT. When invoked, the FATS Label function will write an IBM standard VOL1 label with the specified volume serial and (optionally) an owner name.
- FATS also writes header and trailer labels and tape-marks during the Labeling process, so that the tape is a valid Standard Labeled (SL) Tape. The HDR1/EOF1 labels will contain whatever dataset name appeared on the TAPEX DD statement, unless 'LABEL=EXPDT=98000' is coded, in which case the dataset name will consist of 17 zeros, so that it will be an acceptable *Scratch* tape for Tape Management Systems.
- The dataset Creation Date and Expiration Date will be set to the current date. The HDR2/EOF2 labels will show DCB parameters of RECFM=U and BLKSIZE=32760.
- The tape device type will be determined by the UNIT parameter on the TAPEX DD statement.
- The SUMMPRT DD will receive a summary report, showing the results of the requested function against each tape processed. If SUMMPRT is not coded, the summary will be written to the SYSPRINT dataset.

Tape Certification

FATS certification identifies errors (and their locations on the tape) before an application program has the chance to encounter those errors. FATS carries out this certification by writing large blocks of patterned data on the tape. The hardware readback function of the tape drive is then used to detect blocks that were not written successfully.

For obvious reasons, this certification process is only used for *empty* tapes, or for tapes that contain data no longer required. If existing data is to be checked for ‘readability’, this should be done with the *verification* process—see ‘*Tape Verification*’ later.

When FATS encounters a readback error during the certification process, this is known as a data check. These errors may be caused by a problem with the tape drive equipment, by dirty tape read/write heads, by tape surface damage or improper manufacture, or by dirt or other contaminants on the surface. Equipment failure 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 picks up the dirt and re-deposits it in a different location.

During the certification of a tape, FATS inhibits the operating system and tape hardware from automatically attempting a recovery from data checks, allowing FATS to identify and report on them. When data checks occur, FATS will backspace over the bad block and retry the WRITE. If a user-specified RETRY limit is exceeded (default is 10), the error is considered permanent and is probably due to some tape media defect. If the WRITE is completed successfully before the RETRY limit is reached, it is considered a temporary data check. This may be due to debris on the tape that was dislodged during the retry process.

Here is an example job that will CERTIFY and LABEL a sequence of tapes.

Sample FATS CERTIFY/LABEL job:

```
//CERTLAB EXEC PGM=FATS, REGI ON=OM
//SYSPRI NT DD SYSOUT=*
//SYSPRI N1 DD SYSOUT=*
//SUMMPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1 DD DSN=FATS, UNI T=( TAPE, , DEFER) ,
// LABEL=( , BLP) , DI SP=( , KEEP)
//SYSI N DD *
```

WRITE(1) VOL=(PRD100) , VOLI NCR=1, MAXVOLN=20, MAXERR=5, MAXCERR=10

Some points to note about the above job:

- The ‘WRITE(1)’ parameter tells FATS to certify the tapes that are mounted on the tape drive requested by the TAPE1 DD statement.
- The tapes will be labeled during the certification process, beginning at volser PRD100 and incrementing 1 at a time. The MAXVOLN parameter tells FATS to continue to call for scratch tapes until 20 have been certified and labeled.
- The MAXERR=5 and MAXCERR=10 parameters will cause the certification of any given tape to terminate if more than 5 permanent errors or 10 inches of continuous permanent errors are encountered.
- When certifying tapes that are already labeled, FATS will by default read the existing volume serial from the tape being certified and re-label the tape with it, thus “saving” the original volume serial. This can be overridden if required.

PART TWO

FATS

Once a tape has been certified, FATS will then report on the number and location of any temporary and/or permanent errors found. The number of retries (for temporary errors) and the length of error (for permanent errors) will also be reported. If permanent errors occur at consecutive locations, FATS will report the cumulative length of the error. The locations and lengths of these errors can be reported in either metric or non-metric units.

Here is an extract of the DETAIL report produced from a FATS CERTIFY job, showing the location of temporary and permanent data checks encountered during the certification of tape PRD104. Note the permanent data check at location 104 FT.

F A T S D E T A I L R E P O R T											
MSG	ID	UCB	LBL	OPT	PASS	FILE	REC	LOC	LGTH	RETRIES	ACTION
---	--	---	---	---	----	----	---	---	----	-----	-----
FATS107	TAPE1	383	PRD104	WRITE	1			1 FT			LABEL SAVED
FATS204	TAPE1	383	PRD104	WRITE	1			5 FT		05	TEMP DATA CHECK
FATS204	TAPE1	383	PRD104	WRITE	1			56 FT		06	TEMP DATA CHECK
FATS204	TAPE1	383	PRD104	WRITE	1			67 FT		03	TEMP DATA CHECK
FATS204	TAPE1	383	PRD104	WRITE	1			79 FT		01	TEMP DATA CHECK
FATS204	TAPE1	383	PRD104	WRITE	1			104 FT		05	TEMP DATA CHECK
FATS204	TAPE1	383	PRD104	WRITE	1			104 FT	5 IN	10	PERM DATA CHECK *
FATS204	TAPE1	383	PRD104	WRITE	1			105 FT		03	TEMP DATA CHECK

By receiving a report of the number, length, and location of all errors, it is possible to make an informed decision as to whether each tape certified is of a suitable quality, or if it should be discarded.

The locations of the errors are calculated by FATS in feet or meters from the beginning of the tape. This is done by taking 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 also produces a SUMMARY report at the end of the certification process. Here is an example of the report that is produced.

F A T S S U M M A R Y R E P O R T																
MESSAGE	ID	UCB	LABEL	OPTION	COMP CODE	LENGTH	*** DATACHECKS BY # OF RETRIES ***									
							1	2	3	4	5	6	7	8	9	10
-----	--	---	----	-----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----		
FATS107	TAPE1	383	PRD101	WRITE	NORMAL	00510 FT	4	3	1	0	3	0	0	0	0	
FATS107	TAPE1	383	PRD102	WRITE	NORMAL	00523 FT	2	1	0	1	0	0	0	0	0	
FATS107	TAPE1	383	PRD103	WRITE	NORMAL	00507 FT	0	0	0	0	0	0	0	0	0	
FATS107	TAPE1	383	PRD104	WRITE	NORMAL	00516 FT	1	0	2	0	2	1	0	0	1	

As you can see, there is a line for each of the tape volumes that was certified and labeled, indicating the number of RETRIES that were necessary to overcome temporary data checks on the tape.

Notice that the length of each tape has also been reported. This figure will be as accurate as possible, but potential Inter-Block gap variations on tape/cartridge drives from different vendors could affect its accuracy.

Tape Verification

As mentioned earlier, certification is done on empty tapes, or on tapes that no longer contain valid data. Tape *verification*, on the other hand, is carried out tapes that already contain data and it is used to identify any potential data checks on those tapes that may affect their readability when later processed by an application.

When reading magnetic tape, the operating system or tape control unit will retry any read data check many times by backspacing and re-reading the block. Cleaning actions (*i.e.* 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 will usually abnormally terminate.

FATS read verification can be used on critical or archival tapes *before* they are required by an application job. It can verify the readability of these tapes to ensure that application read failures are pre-empted.

- FATS performs the tape verification by reading data blocks from the tape *without* actually transferring it to the CPU. This is sufficient to identify data checks on the tape without using expensive CPU resources to do it.
- FATS will read each block and attempt to re-read any encountered data checks a user-specified number of times. The default is 10, but this can be increased. Often, a sufficient number of retries will overcome temporary data checks.
- FATS can also verify that a tape has been properly formatted. It does this by checking each block against the tape DCB.

After the verification, the FATS DETAIL report (shown earlier) indicates the exact location, file number and record number(s) of any temporary and/or permanent read data checks encountered. Again, FATS will report the length of any permanent errors and the total length of any consecutive permanent errors.

The following sample JCL illustrates the concurrent verification of *two* specific tapes, described by the TAPE1 and TAPE2 DD's.

Sample FATS VERIFY job:

```
//VERI FY    EXEC  PGM=FATS, REGI ON=OM
//SYSPRI NT  DD    SYSOUT=*
//SYSPRI N1  DD    SYSOUT=*
//SUMMPRT   DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPE1      DD    DSN=DATA. SET. NAME1, UNI T=TAPE,
//           VOL=SER=PRD010, DI SP=OLD
//TAPE2      DD    DSN=DATA. SET. NAME2, UNI T=TAPE,
//           VOL=SER=PRD099, DI SP=OLD
//SYSI N     DD    *
```

```
READ( 1)
READ( 2)
```

Note: The 'read verification' carried out by FATS is sufficient to identify data checks on a tape. However, for a more detailed look at the data causing the errors, a FATAR ANALYZE may be executed. FATAR is also the more appropriate tool to use for verifying tapes where the actual number of data files on the tape is unknown. See PART THREE for more details.

Tape Erasure

FATS also provides a facility to ERASE the contents of a tape. This is often necessary when:

- *An installation is upgrading to new tape technology.* The company will probably sell-off or discard many of their old tapes, so the data contained on these tapes, which may be company confidential, should be completely erased.
- *Existing tapes are being sent offsite.* For security reasons, the 'old' data beyond the current end-of-tape marker should be completely erased.

All tape drives support a feature called “*Data Security Erase*”, which can be used to erase data on a tape at high-speed without tying up the tape control unit. The erase can be done without impacting other jobs that are using tape drives on the same tape control unit. The erase will take place at maximum tape write speed (which varies by tape drive model). The “*Data Security Erase*” is not supported by any standard IBM software, but FATS fully supports this function.

By default, a FATS ERASE will preserve the existing volume serial on the tape. Alternatively, it can either be re-labeled with a new serial number, or an unlabeled tape can be created. The hardware erase function will then be started and will continue without intervention from the CPU or control unit until the end-of-tape is reached. For tape cartridge systems, this involves writing a random pattern on the tape for complete security. For round tapes, the tape drive's erase head is used to erase the tape to “unrecorded” status.

Here is an example of a FATS ERASE job:

Sample FATS ERASE job:

```
//ERASE EXEC PGM=FATS, REGI ON=OM
//SYSPRI NT DD SYSOUT=*
//SYSPRI N1 DD SYSOUT=*
//SYSPRI N2 DD SYSOUT=*
//SYSPRI N3 DD SYSOUT=*
//SUMMPRT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//TAPE1 DD DSN=DUMMY, UNI T=(TAPE, , DEFER) ,
// LABEL=(, BLP), DI SP=(, KEEP)
//TAPE2 DD DSN=DUMMY, UNI T=(TAPE, , DEFER) ,
// LABEL=(, BLP), DI SP=(, KEEP)
//TAPE3 DD DSN=DUMMY, UNI T=(TAPE, , DEFER) ,
// LABEL=(, BLP), DI SP=(, KEEP)
//SYSI N DD *

DEFAULT VOLINCR=1, MAXVOLN=50
ERASE(1) VOL=100001
ERASE(2) VOL=100051
ERASE(3) VOL=100101
```

In this example, 150 tapes are to be completely erased. They have consecutive volsers (100001—100150), so the 'VOL=', 'VOLINCR=' and 'MAXVOLN=' parameters are being used to erase and re-label 50 tapes on each of the three drives available.

FATS will automatically call for the tapes by volser as needed. Although the 'LABEL=(,BLP)' parameter has been coded to suppress operating system label handling, FATS will re-label each tape as it erases it.

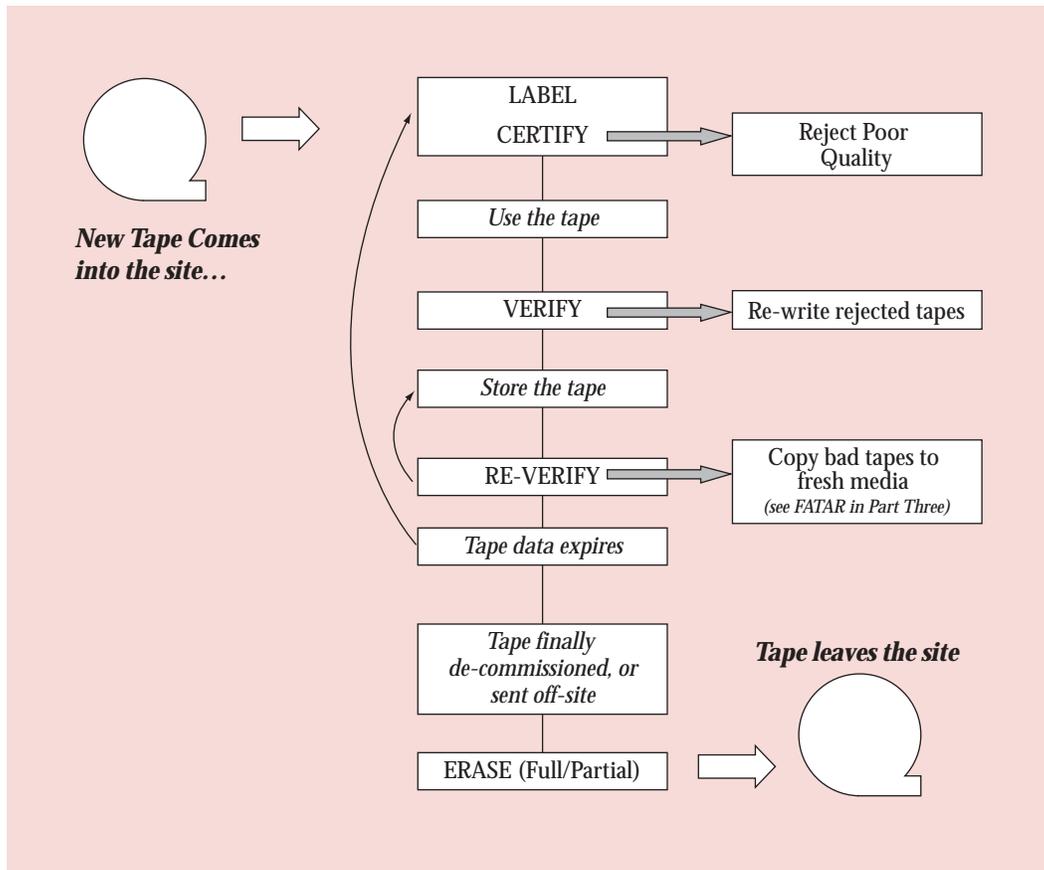
The above example shows the *whole* tape being erased. It should be noted, however, that FATS also allows for the *partial* erasure of tapes. For example, the 'new' data can be written on top of an old tape containing data and then FATS (via the NOREWIND option) can be used in a subsequent step in the same job to erase the remaining tape. This is far more efficient than erasing the whole tape first and then re-winding the tape before writing the new data.

PART TWO

FATS

Summary

The LABEL, CERTIFY, VERIFY and ERASE functions of FATS can be used to form the basis of a highly effective 'Tape Media Quality' program:



The Tape Media Quality Program:

- New tapes are immediately **certified** as soon as they are brought into the tape pool, at the same time as they are **labeled**. Tapes that are identified as having media problems can be rejected before any data is written to them.
- If critical data is written to the tape, the readability of this data can then be **verified**. Any problems encountered with the data can be corrected before it is required by the application.
- Any data that is to be stored on a long-term basis (e.g. backups, financial data, etc) can be subjected to a program of repeated **verification** to ensure the continued readability of the data does not get affected as a result of the long-term storage of the media.
- When the data on a tape reaches the end of its retention period, the tape can then be **re-certified** to ensure the continued quality of the tape media.
- Finally, when a tape is de-commissioned from the site, or is sent offsite containing data, either a full or partial **erase** of the tape can be used to ensure that the security of company confidential data is maintained.

PART THREE

FATAR & FATSCOPY

Introduction

The FATAR and FATSCOPY functions described in PART ONE can be executed by running a batch job under z/OS or OS/390. This section of the guide shows some examples of those batch jobs. PART FOUR shows how the same functions can be requested through a supplied set of ISPF panels.

Mapping Tapes

Tapes with unknown characteristics (volser, filenames, etc) are virtually impossible to deal with. They may arrive at the site with insufficient documentation, or the creating job may have been lost. Consider the following questions:

- What data format does the tape contain—EBCDIC or ASCII?
- Is the tape Labeled or Unlabeled. If labeled, what is the label type—IBM, ISO/ANSI?
- What files are on the tape? How Many? What names? What DCB Characteristics?
- Was the data written with Hardware Compression (IDRC)?

These questions can be easily answered by running a FATAR ANALYZE against the tape. The following job illustrates a simple ANALYZE to map out the contents of an unknown tape.

FATAR—Map an unknown tape:

```
//FATAR      EXEC  PGM=FATAR, REGION=OM
//SYSPRINT   DD   SYSOUT=*
//TAPESUMM   DD   SYSOUT=*
//TAPEIN     DD   DSN=FIRSTDS, UNIT=TAPE, VOL=SER=999999,
//           LABEL=(, BLP, EXPDT=98000), DISP=OLD
//SYSIN      DD   *
```

ANALYZE

Some points to note about the above job:

- LABEL=(,BLP) is specified on the TAPEIN DD Statement so that any tape can be mounted when the job issues a mount for volser 999999. The 'DSN=FIRSTDS' is just a dummy value to conform to JCL requirements.
- The EXPDT=98000 has been specified so that the Tape Management System does not attempt to verify volume 999999.
- 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.

At the end of the job, FATAR will print a summary of the files and data on the tape. This summary will be printed to the TAPESUMM DD and will look like this:

FATAR TAPE SUMMARY FOR TAPE VOLUME -800432- AT DENSITY 38000 BPI 07/10/03																
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	RAY.SCR010	800432	0001	1986/192	00/000	U	00000	SPSN	0	486	17M	0	24	34213	48092	39
			0001				32760	COPY1				0				
5	RAY.SCR501	800432	0002	1986/192	00/000	U	00000	SPSM	0	580	24M	0	24	40931	49060	56
			0001				32760	COPY1				0				
				HIGHEST EXPIRATION ==>>>		00/000		TOTALS ==>>>		1066		40M				95

PART THREE

FATAR & FATSCOPY

FATAR will also print out (to SYSPRINT) a DETAIL report like the one shown here:

```
FATS014 NO CONTROL CARD DATASET. DEFAULTS ASSUMED
FATS071 TAPE BUFFER SIZE IS 65535 BYTES

                                CHARACTERISTICS OF THE TAPE TO BE ANALYZED
                                UNIT DEVICE SERIAL
                                0391 3480XF 800432

                                FATAR DETAIL REPORT

BLOCK  LNPTH/  MESSAGE/          1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80
NUMBER DISPL  BLOCK TYPE          (COLUMN GRID IS VALID ONLY FOR CHARACTER FORMATTED DATA)

* * * * * START FILE      1H
  1  80  INPUT LABEL          VOL1800432
      ***VOLUME LABEL***
      VOLSER=800432 OWNER=
  2  80  INPUT LABEL          HDR1RAY.SCR010          80043200010001          86192 000000000000IBM OS/V5 370
      ***HEADER LABEL***
      DSNNAME=RAY.SCR010          SERIAL=800432 VOLSEQ#=0001 FILESEQ#=0001
      CRTDT= 98192 EXPDT= 00000 SECURITY=0 BLOCKCOUNT= 000000
  3  80  INPUT LABEL          HDR2U32760000000SPSN /COPY1          14850
      ***HEADER LABEL***
      RECFM=U LRECL=00000 BLKSIZE=32760 BUFOFF(ANSI)=
      DENSITY=0 TRTCH= CREATING JOB/STEP=SPSN /COPY1

* * * * * END OF FILE      1H -- FILE CONTAINED          3 BLOCKS

* * * * * START FILE      1
* * * * * END OF FILE      1 -- FILE CONTAINED          486 BLOCKS

* * * * * START FILE      1T
  1  80  INPUT LABEL          EOF1RAY.SCR010          80043200010001          86192 000000000486IBM OS/V5 370
      ***END-OF-FILE***
      DSNNAME=RAY.SCR010          SERIAL=800432 VOLSEQ#=0001 FILESEQ#=0001
      CRTDT= 98192 EXPDT= 00000 SECURITY=0 BLOCKCOUNT= 000486
  2  80  INPUT LABEL          EOF2U32760000000SPSN /COPY1          14850
      ***END-OF-FILE***
      RECFM=U LRECL=00000 BLKSIZE=32760 BUFOFF(ANSI)=
      DENSITY=0 TRTCH= CREATING JOB/STEP=SPSN /COPY1

* * * * * END OF FILE      1T -- FILE CONTAINED          2 BLOCKS

* * * * * START FILE      2H
  1  80  INPUT LABEL          HDR1RAY.SCR501          80043200010002          86192 000000000000IBM OS/V5 370
      ***HEADER LABEL***
      DSNNAME=RAY.SCR501          SERIAL=800432 VOLSEQ#=0001 FILESEQ#=0002
      CRTDT= 98192 EXPDT= 00000 SECURITY=0 BLOCKCOUNT= 000000
  2  80  INPUT LABEL          HDR2U32760000000SPSM /COPY1          14850
      ***HEADER LABEL***
      RECFM=U LRECL=00000 BLKSIZE=32760 BUFOFF(ANSI)=
      DENSITY=0 TRTCH= CREATING JOB/STEP=SPSM /COPY1

* * * * * END OF FILE      2H -- FILE CONTAINED          2 BLOCKS

* * * * * START FILE      2
* * * * * END OF FILE      2 -- FILE CONTAINED          580 BLOCKS

* * * * * START FILE      2T
  1  80  INPUT LABEL          EOF1RAY.SCR501          80043200010002          86192 000000000580IBM OS/V5 370
      ***END-OF-FILE***
      DSNNAME=RAY.SCR501          SERIAL=800432 VOLSEQ#=0001 FILESEQ#=0002
      CRTDT= 98192 EXPDT= 00000 SECURITY=0 BLOCKCOUNT= 000580
  2  80  INPUT LABEL          EOF2U32760000000SPSM /COPY1          14850
      ***END-OF-FILE***
      RECFM=U LRECL=00000 BLKSIZE=32760 BUFOFF(ANSI)=
      DENSITY=0 TRTCH= CREATING JOB/STEP=SPSM /COPY1

* * * * * END OF FILE      2T -- FILE CONTAINED          2 BLOCKS
* * * * * START FILE      3H
* * * * * END OF FILE      3H -- FILE CONTAINED          0 BLOCKS

FATS020 ANALYSIS TERMINATED AT TAPEMARK SEQUENCE

                                FINAL TOTALS

FILES READ      BLOCKS READ      BYTES READ      FEET READ      TEMP ERRS      PERM ERRS      FILES WRITTEN      BLOCKS WRITTEN
  2              1066              40367286              96              0              0              0              0
```

As you can see, the DETAIL report provides complete information on the two files that were found on the tape (RAY.SCR010 and RAY.SCR501), as well as the header and trailer labels surrounding them.

For each of the files, information such as the Creation Date, Expiration Date, RECFM, LRECL, BLKSIZE has been displayed, together with details of the Job/Step that created the files and the number of data blocks contained within them.

The Final Totals section at the bottom of the report summarizes the information and also indicates that there were no temporary or permanent errors encountered while reading this particular tape.

PART THREE

FATAR & FATSCOPY

In this next example, a standard labeled tape is to be mapped. It is being opened as a labeled tape (no LABEL parameter on the TAPEIN DD), so the volser and first dsn must be specified and the Tape Management System may verify these values. The LENCHK parameter causes FATAR to print the beginning of any block that differs in length from the preceding block (for checking fixed length files) and LBLPRT=DUMP causes labels to be printed in dump format (hex and EBCDIC). The PRINT statement causes blocks 1 to 3 of every data file to be printed.

FATAR—Map an SL tape:

```
//FATAR      EXEC  PGM=FATAR, REGI ON=OM
//SYSPRI NT  DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEI N    DD    SSN=FI RSTDS, UNI T=TAPE, VOL=SER=111111,
//          DI SP=OLD
//SYSI N     DD    *
```

```
ANALYZE  LENCHK, LBLPRT=DUMP
PRINT  LF=ALL, B=1- 3
```

In the following example, FATAR is going to ANALYZE just the 4th and 5th files on a standard labeled tape. The LABEL=4 parameter on the TAPEIN DD tells FATAR to begin with that dataset, and NUMFILES=2 on the ANALYZE tells FATAR to stop after 2 files. The PRINT ALL control statement directs FATAR to print all data blocks (first 200 bytes).

FATAR—Partial Map of an SL tape:

```
//FATAR      EXEC  PGM=FATAR, REGI ON=OM
//SYSPRI NT  DD    SYSOUT=*
//TAPESUMM   DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEI N    DD    DSN=FOURTHDS, UNI T=TAPE, VOL=SER=111111,
//          LABEL=4, DI SP=OLD
//SYSI N     DD    *
```

```
ANALYZE  NUMFILES=2, PRTLEN=200
PRINT  ALL
```

In this last example, a multi-volume tape set containing more than one dataset will be mapped. FATAR will give total block counts for each dataset, even when it crosses volumes. An intermediate block count is given each time an end-of-volume is reached. The tape summary will contain one page for each tape, detailing the files and block counts on it. File numbers will be logical files from the beginning of the first tape, counting files which cross multiple volumes as one file. 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—Map of multi-volume tape set:

```
//FATAR      EXEC  PGM=FATAR, REGI ON=OM
//SYSPRI NT  DD    SYSOUT=*
//TAPESUMM   DD    SYSOUT=*
//TAPEI N    DD    DSN=UNKNOWN, UNI T=TAPE,
//          VOL=SER=(333333, 444444, 555555),
//          LABEL=(, BLP, EXPDT=98000), DI SP=OLD
//SYSI N     DD    *
```

```
ANALYZE
SCAN  LF=2, B=ALL, L=4, P=1, T=Z
```

PART THREE

FATAR & FATSCOPY

Copying Tapes—Basic Features

Copying tapes with standard utilities is often difficult and tedious, particularly multi-file, multi-volume tapes. FATAR provides various mechanisms for copying data from one tape to another. It is impossible within the scope of this document to show all of the possible options. Instead, we will illustrate the most common uses of the copying functions. For a complete detailed description of FATAR tape copying, please see the FATS/FATAR User Guide.

IMAGE COPY

In this first example, FATAR is being invoked to create an image copy of a single tape volume. All of the files on the input tape will be copied to the output tape, together with the input tape's volser.

FATAR—Image Copy:

```
//FATAR      EXEC  PGM=FATAR, REGION=OM
//SYSPRI NT  DD    SYSOUT=*
//TAPESUMM   DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEIN     DD    DSN=ANYDSN, UNIT=3490, VOL=SER=111111,
//           LABEL=(, BLP), DISP=OLD
//TAPEOUT    DD    DSN=NOT. USED, UNIT=3490,
//           LABEL=(, BLP), DISP=(NEW, KEEP)
//SYSIN      DD    *
```

ANALYZE LABELS=IMAGE

Some points to note about the above example:

- You will notice that the JCL and control statements look very similar to the previous examples for mapping tapes. The only changes are the addition of LABELS=IMAGE on the ANALYZE statement and the added TAPEOUT DD statement.
- All of the data on the input tape (TAPEIN) will be copied across to the output tape (TAPEOUT), including all the labels and the volume serial, creating an exact bit-for-bit copy.
- Since the output tape will have the same volume serial as the original, tape management systems and system catalogs will not have to be modified or updated.
- LABEL=(,BLP) is coded on the TAPEOUT DD statement so that all labels and data can be copied to the output tape. LABEL=(,BLP) is optional on TAPEIN.
- DETAIL and SUMMARY reports like the ones shown earlier will be produced for the input tape.
- If the input tape was written with IDRC compaction, additional information will be included in the DETAIL and SUMMARY reports showing the savings that were achieved. In the following extract from a SUMMARY report, the data on the input tape had been written with IDRC compaction, producing a saving of 48%.

FATS020 ANALYSIS TERMINATED AT TAPEMARK SEQUENCE

FINAL TOTALS

FILES READ	BLKS READ	BYTES READ	METERS READ	TEMP ERRS	PERM ERRS	FILES WRITTEN	BLKS WRITTEN	COMPACTED	%USED
1	21127	83081159	****	0	0	1	21127		
IDRC -#	136	43470848	11					48%	5%

PART THREE

FATAR & FATSCOPY

PARTIAL COPY

In the following job, a partial copy of a tape will be performed. Files 3 and 4 will be copied from labeled tape 111111 and written to a scratch Magstar (3590). The LABEL=3 parameter on the TAPEIN DD Statement causes FATAR to start the copy with file 3 on the tape and NUMFILES=2 limits the copying to just two files.

The files will become files 1 and 2 on the output tape. The RENAME statement is used to specify new names for the 2 output files, and CAT=YES will cause FATAR to catalog those names to the output tape.

COMPACTION=COMP will cause the output tape to be written using IDRC hardware compaction. REBLOCK=16000 will cause each copied file to be reblocked to as close to a blocksize of 16000 bytes as possible.

FATAR—Partial Copy:

```
//FATAR          EXEC  PGM=FATAR, REGI ON=OM
//SYSPRI NT      DD    SYSOUT=*
//TAPESUMM       DD    SYSOUT=*
//SYSUDUMP       DD    SYSOUT=*
//TAPEIN        DD    DSN=THI RDDS, UNI T=TAPE, VOL=SER=111111,
//              LABEL=3, DI SP=OLD
//TAPEOUT        DD    DSN=DUMMY, UNI T=3590- 1, DI SP=(NEW, KEEP)
//SYSI N         DD    *
```

```
ANALYZE
NUMFILES=2, REBLOCK=16000, COMPACTION=COMP, CAT=YES
RENAME  LF=3, NEWN=TECHSERV. FILE3
RENAME  LF=4, NEWN=TECHSERV. FILE4
```

LABEL CONVERSION

In this final example, file 3 from standard label (SL) tape 111111 is to be copied to an unlabeled (NL) scratch tape. LABEL=3 on the TAPEIN DD statement positions FATAR to the proper file. The NUMFILES=1 parameter restricts FATAR to copying one file.

LABEL=(,NL) on the TAPEOUT DD statement requests an unlabeled scratch and causes FATAR to copy the TAPEIN data without labels. The input dataset name (from the TAPEIN DD statement) is used to name the output file.

FATAR—Copy SL to NL:

```
//FATAR          EXEC  PGM=FATAR, REGI ON=OM
//SYSPRI NT      DD    SYSOUT=*
//TAPESUMM       DD    SYSOUT=*
//SYSUDUMP       DD    SYSOUT=*
//TAPEIN        DD    DSN=THI RDDS, UNI T=TAPE, VOL=SER=111111,
//              LABEL=3, DI SP=OLD
//TAPEOUT        DD    DSN=DUMMY, UNI T=TAPE,
//              LABEL=(, NL), DI SP=(NEW, KEEP)
//SYSI N         DD    *
```

```
ANALYZE  NUMFILES=1
```

FATAR & FATSCOPY

Copying Tapes—FATSCOPY

The basic copying features of FATAR are adequate for copying small numbers of files or tapes, but if a larger amount of copying is to be done, as part of a “media conversion” or “tape stacking” exercise, the FATSCOPY cost-option feature of FATAR automates the process. Tape files are located via the MVS catalogs and can be selected for copying by the following criteria:

- File name—fully qualified or masked.
- Tape volume and/or device type to which they are currently cataloged.
- The File Sequence numbers (fileseq) that the files appear on the tape.
- Creation Date, Expiration Date or Size (in MB) of the tape files.
- The name of the job, step or program that created the tape files.
- Files cataloged to EMC CopyCross virtual Volumes.

Note: Backup tape files belonging to FDRABR, and backup/migration tape files belonging to DFSMSHsm will be automatically excluded by matching the tape dataset name against various masks.

A *simulation* feature makes it easy to verify that the expected files will be selected for copying before the process is run for real. The capacity of the output tapes can be specified and the simulation will estimate the number of volumes required to hold the tape files to be copied.

Once selected, the input datasets are dynamically allocated by FATSCOPY and then copied to consecutive files on the output tape. Files from multiple tapes, but with like expiry dates, can be placed onto the same output tape. Information recorded in the Tape Management System (FATSCOPY currently supports CA-1 and DFSMSrmm) can be propagated to the tape management records of the output datasets, effectively re-cataloging the files to the output tape. The input tapes can optionally be returned to scratch status for reuse.

Let’s take a look now at the two most common uses of the FATSCOPY function.

PART THREE

FATAR & FATSCOPY

TAPE MEDIA CONVERSION

FATSCOPY is an ideal tool for migrating from one type of tape device to another. When installing higher capacity drives, a percentage of tape data will automatically get migrated as new datasets are created and old ones expire. There will, however, be a proportion of long-term retention tape data that will remain static on the old tapes unless it is specifically migrated. FATSCOPY can be used to copy this residual data to the new tapes. The following example job illustrates how easy it is to use FATSCOPY to copy some tape datasets from one media to another.

FATSCOPY—Media Conversion:

```
//CONVERT      EXEC    PGM=FATSCOPY, REGI ON=OM
//SYSPRI NT    DD      SYSOUT=*
//SELRP T      DD      SYSOUT=*
//COPYRPT     DD      SYSOUT=*
//TAPESUMM    DD      SYSOUT=*
//SYSUDUMP    DD      SYSOUT=*
//TAPEOUT     DD      DSN=OUTPUT, UNI T=STK9840, DI SP=(, KEEP)
//SYSI N      DD      *

COPY          TMSI NPUT=SCRATCH, MULTI FILE=YES
SELECT       CATDSN=**, DEVTYPE=3490
```

All of the tape datasets (CATDSN=**) that are currently cataloged to the 3490E tape device type will be copied and stacked onto StorageTek 9840 cartridges. The esoteric name of 'STK9840' has been defined in this installation to allocate a 9840 drive. Since this is a migration, MULTIFILE=YES has been coded to allow tapes already containing multiple files to be copied. Defaults have been taken to ensure that the copied files are recataloged to the output tape and the input files are returned to scratch status. Lists of the files selected and copied are written to the SELRPT and COPYRPT DD's.

PART THREE

FATAR & FATSCOPY

TAPE STACKING

As we highlighted in PART ONE, tape technology continues to be improved. Every year or so a new generation of tape drives and tape media is introduced, allowing users to put more data onto the same size tape cartridge.

Unfortunately, many tape datasets are small or vary in size from one run to another and few tape applications are designed to place more than one dataset on a tape volume. The result is that many tapes in a tape library contain only a fraction of the data they are capable of holding. This results in inefficient use of the physical space in the library, which is a potentially expensive problem if an automated tape library (ATL/silo) is involved. It also requires more cartridges, which is an additional expense.

These problems can be minimized and the resulting expenses avoided by putting more data on each cartridge—known commonly as ‘tape stacking’. Unfortunately, although it is possible to automate dataset stacking by changing application JCL, it can be labor-intensive to make the required changes and to monitor and correct them as application needs change.

FATSCOPY avoids these problems and automates the tape stacking process by working with datasets that have already been created on tape. The applications are left to run without modification, creating datasets on tape as they always have. Then, periodically, FATSCOPY is run to collect all of these inefficient tape datasets and stack them on to fewer tape volumes, releasing the input volumes as scratch tapes for reuse.

As with the ‘media conversion’ described above, the stacked datasets can be re-cataloged so that they can be automatically located by the applications if and when required. The stacking of tape datasets is, in effect, transparent to the applications.

The following example job illustrates how easy it is to use FATSCOPY to perform this type of tape dataset stacking.

FATSCOPY—Tape Stacking:

```
//STACK      EXEC  PGM=FATSCOPY, REGI ON=OM
//SYSPRI NT  DD    SYSOUT=*
//SELRPT     DD    SYSOUT=*
//COPYRPT   DD    SYSOUT=*
//TAPESUMM  DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEOUT   DD    DSN=OUTPUT, UNI T=3590-1, DI SP=(, KEEP)
//SYSI N     DD    *

COPY
SELECT    CATDSN=APPL%. *. SAVE. **,
           CRDATE>=2000. 001, CRDATE<=2000. 100
```

This example is similar to the ‘media conversion’ example shown on the previous page. On this occasion, datasets belonging to a single application are being selected by FATSCOPY’s powerful and flexible name masking feature.

In this particular case, we are looking for all datasets that have a five-character high-level qualifier, of which the first four characters are ‘APPL’ and the fifth character is any single valid alphanumeric or national character. The datasets can then have any second level, but the third level must be ‘SAVE’. There can then be any number of additional levels in the name.

Only datasets matching the name mask, and which were created between 2000.001 and 2000.100, will be selected for the stacking process. They will be copied and stacked onto an IBM 3590 cartridge. Once again, the copied files will be recataloged to the output tape, and the input tapes will be returned to scratch status.

PART THREE

FATAR & FATSCOPY

Another useful feature of FATAR is its ability to recovery data from tapes that have been **PARTIALLY OVERWRITTEN**. In this next example, a cartridge tape containing valid data was accidentally used for output, but the new file is smaller than the original file. FATAR will be used to recover as much of the original data file as possible.

FATAR—Recover From Partial Overwrite:

```
//FATAR      EXEC  PGM=FATAR, REGI ON=OM
//SYSPRI NT  DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEI N    DD    DSN=ORI GI NAL, UNI T=TAPE,
//          VOL=SER=BBBBBB, DI SP=OLD,
//          DCB=( RECFM=FB, LRECL=125, BLKSI ZE=32750)
//TAPEOUT    DD    DSN=ORI GI NAL, UNI T=TAPE, DI SP=(NEW, KEEP)
//SYSI N     DD    *
```

ANALYZE LABELS=EOD, CAT=RECAT, BLP

Some points to note about the above example:

- All cartridge drives (except the 3480) write an EOD mark (End of Data) after the last block written to a tape. When a tape is partially overwritten, the remainder of the previous data is beyond the EOD. LABELS=EOD causes FATAR to search for the first hardware EOD mark on the tape and recover data that follows it.
- Since header labels are not available for the overwritten file, the DCB characteristics of the original data should be provided on TAPEIN, if known. The TAPEOUT DD statement specifies a labeled scratch tape with the dataset name of the original dataset.
- The first data block is likely to cause a permanent data check if it is partially overwritten. If this is the only error on the tape it can be ignored.
- CAT=RECAT will cause the dataset to be re-cataloged to the output tape. The correct dataset name will have to be specified on the TAPEIN DD statement. If there are additional original files beyond the overlaid file, FATAR will automatically recover them as well, using the dataset name and DCB information in their header labels to copy and catalog them to the output tape.

This next example shows a similar recovery for 3480 cartridges (or 3420 tapes), which do not write an EOD. The new file that has caused the overwrite will occupy four physical files on the tape (header labels, data, trailer labels, and an 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.

FATAR—Recover From Partial Overwrite:

```
//FATAR      EXEC  PGM=FATAR, REGI ON=OM
//SYSPRI NT  DD    SYSOUT=*
//SYSUDUMP   DD    SYSOUT=*
//TAPEI N    DD    DSN=ORI GI NAL, UNI T=TAPE,
//          VOL=SER=BBBBBB, LABEL=( 5, BLP) , DI SP=OLD,
//          DCB=( RECFM=FB, LRECL=125, BLKSI ZE=32750)
//TAPEOUT    DD    DSN=ORI GI NAL, UNI T=TAPE, DI SP=(NEW, KEEP)
//SYSI N     DD    *
```

ANALYZE CAT=RECAT

PART THREE

FATAR & FATSCOPY

Another common problem that FATAR can be used to recover from is an **IMPROPERLY CLOSED TAPE**. Due to a system crash or power failure, tape 333333 in the following example was not properly closed and has no trailing tape mark or labels and it cannot be read with standard access methods. A previous FATAR ANALYZE job showed that block 742 is the last good block of the proper length and a data check occurs immediately after it. The following FATAR COPY job will copy the tape up to and including block 742 onto a scratch tape with proper labels.

FATAR—Recover From Improper Closure:

```
//FATAR          EXEC  PGM=FATAR, REGION=OM
//SYSPRI NT      DD    SYSOUT=*
//SYSUDUMP       DD    SYSOUT=*
//TAPEI N        DD    DSN=A. NAME, UNIT=TAPE,
//              VOL=SER=333333, DISP=OLD
//TAPEOUT        DD    DSN=DUMMY, UNIT=TAPE, DISP=(NEW, KEEP)
//SYSI N         DD    *
```

ANALYZE ENDAFTER=(LF=1, B=742)

Summary

This section has given some insight into the various capabilities within FATAR and FATSCOPY. We have concentrated on some of the more common uses of the product, but they can also offer other facilities that prove useful from time-to-time.

For example, some features that we have not been able to cover in this guide include:

- Converting data type (e.g. ASCII-EBCDIC)
- Dropping blocks/records during a tape copy
- Altering the contents of block/records during the copy
- Detecting and correcting variable spanned records
- Verifying that certain data fields contain valid data
- Scanning a tape file for a data match

More information on these features can be found in the “FATS/FATAR & FATSCOPY” product manual.

PART FOUR

THE ISPF PANELS

Introduction

So far in the Guide, we have illustrated the use of FATS/FATAR through the running of batch jobs. The programs can also be invoked through a full set of ISPF panels that are provided with the product.

Note: The copying/stacking features of FATSCOPY, as described in Part 3, are only available as a batch process.

All of the FATS/FATAR functions that we have discussed in this Guide are available through the panels, which can be tailored so that only certain functions are provided to certain user groups. It would be possible, for example, to allow inexperienced Operations personnel the ability to Label tapes through the panels, while keeping the panels for the more specialized functions (FATS Erase, FATAR Error Recovery, etc) restricted to experienced personnel.

The Main Menu

The Main Menu of the FATS/FATAR ISPF panels looks like this:

```
----- FATS/FATAR  MAIN  MENU -----
OPTION ==>
A.  CERTIFY    new tapes
B.  CERTIFY    existing labeled tapes whose datasets 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
O.  RECOVER    data from overwritten (damaged) tape
Q.  QUERY      Tape Management System - CA-1/TMS (V5.1)
S.  SETUP      JCL and FATS/FATAR defaults for user           (Profiles)
```

Options A-F are the Certify, Label, Erase and Verify functions of FATS, as described in PART TWO of this Guide. Options G-O are the various functions of FATAR that were described in PART THREE of this Guide, such as Mapping tapes, Copying tapes and Recovering from tape errors.

For all of the options A-O, a second panel will be displayed. Once the relevant information has been supplied on that panel, FATS or FATAR JCL and control statements will be created as jobstreams that can be edited prior to submission, or saved for later reuse.

If the FATS/FATAR Tape Management Interface has been installed, option Q will also appear on this menu. As shown above, it identifies the type and version of Tape Management System that FATAR has identified as being active on the system. Option Q allows the Tape Management System's Database to be queried through facilities provided by FATS/FATAR.

The final option (Option S) leads to a series of panels where defaults can be set for the operation of the FATS/FATAR panels. These defaults include JCL options for the generation of JOB statement and other JCL parameters. FATS/FATAR "profile" options can also be set, giving values for operands, such as lines-per-page, which cannot be overridden on other panels. There are also some other FATS/FATAR operands that *can* be overridden on other panels, but defaults can be changed here to save time later.

PART FOUR

THE ISPF PANELS

Sample FATS Panels

Selecting Option A from the Main Menu would bring up the panel displayed below, which can be used to Certify and Label tapes.

This panel would normally be used to create IBM standard labeled tapes, but unlabeled tapes or ANSI labeled (AL) tapes can also be created by specifying YES to either the NOLABEL or ANSI option at the bottom of the panel.

BLP was coded in the 'Label' field because the tapes being certified and labeled were brand-new and needed to be mounted without trying to read existing labels.

```
----- FATS CERTIFY NEW TAPES -----
COMMAND ===>                                     Enter Q for TMS Query
TAPE1 DD Dsname = BAB.FATSOUT.JOBT                <
        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)
ANSI     - Create ANSI labeled tapes : NO (yes/no)
Display more FATS Operands?         : NO (yes/no)
```

When labeling tapes, the volume serials can be specified in one of two ways:

- If the tapes are to be assigned a selection of random volsers, up to 20 specific volsers can be specified in the fields provided.
- If the tapes are to be labeled in sequence, the first volser can be specified in field 1, together with an appropriate value coded in the MAXVOLN field.

Panels similar to the one shown above are also available to CERTIFY existing tapes (option B), LABEL new tapes (option C), ERASE tapes (option E) and VERIFY tapes (option F).

As an example to illustrate the similarity of the various panels, the one displayed below is the one used for erasing tapes (option E). As you can see, the parameters are the same as for the panel shown above, except for the SAVLAB option that allows the existing volume serials to be preserved after the erase.

```
----- FATS ERASE OLD DATA -----
COMMAND ===>                                     Enter Q for TMS Query
TAPE1 DD Dsname = BAB.FATSOUT.JOBT                <
        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? s : YES (yes/no)
NOLABEL - Create no label (NL) tapes : NO (yes/no)
ANSI     - Create ANSI labeled tapes : NO (yes/no)
Display more FATS Operands?         : NO (yes/no)
```

PART FOUR

THE ISPF PANELS

Sample FATAR Panels

A similar set of panels exists for the FATAR functions (Options G to O). The panel shown below is the one that is displayed for Option H to EXAMINE a tape (which invokes a FATAR ANALYZE).

If the default of BLP is used, the dataset name and volume serial are not actually verified against the tape, so any values required to get the proper tape mounted and opened can be used. This allows any tape, including tapes with unknown contents and those from outside the installation to be mounted.

Changing the label type to SL or AL requires the actual dataset name and volume serial to be entered. Multiple volume serials may be entered if they are part of a multi-volume tape set. FATAR will report on the actual volume serial, label type, dataset names, and contents of every file on the tape. The PRINTOPT options allow you to print data from the tapes, or to print only the labels from the first file on the tape (to quickly identify the contents of tapes).

```
----- FATAR EXAMINE (MAP) TAPES -----
COMMAND ==>                                     Enter Q for TMS Query
TAPEIN DD Dsname=  BAB.FATARIN.JOBT                <
          Unit  =( TAPE          )
          Label =( ,BLP,EXPDT=98000          )
          Volser=( 99999T                )

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)
          (A)- Print ALL blocks in ALL files    : NO      (yes/no)
          (B)- Number of blocks to print for each file processed: _____

Display more FATS Operands?                       : NO      (yes/no)
```

The following panel is the one displayed for the 'Copy a tape volume' (Option J), which generates a FATAR job to make an image copy (an exact bit-for-bit copy) of a single tape volume. The submitted job must be authorized to use BLP (Bypass Label Processing) since this function is required for image copy.

If INPUT or ALL is specified for VALIDATE, the volume serial of the input tape will be checked against the serial provided. The dsname of TAPEOUT will not be used, and the TAPEOUT volume serial is whatever is required to get the proper output tape mounted. The output tape will have the actual volume serial of the input tape after the copy is complete, unless YES is specified for OUTSER.

```
----- FATAR IMAGE COPY -----
COMMAND ==>                                     Enter Q, QI or QO for TMS Query
TAPEIN  DD Dsname=  BAB.FATARIN.JOBT                <
          Unit  =( TAPE          ),Label=( ,BLP,EXPDT=98000          )
          Volser=( 99999T                )
TAPEOUT DD Dsname = BAB.FATAROUT.JOBT                <
          Unit  =( TAPE          ),Label=( ,BLP,EXPDT=98000          )
          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)
OUTDSN    - Check for initialized Output Dsname  : NO      (yes/no)
OUTVOL    - Use Input Volser for Mount/Validation: NO      (yes/no)
OUTSER    - Retain Output Volume Serial Number  : NO      (yes/no)
COMPACTION- Data compression requested for Output: SYSTEM (system/copy/yes/no)

Display more FATS Operands?                       : NO      (yes/no)
```

PART FOUR

THE ISPF PANELS

The following panel is the one used for recovering data from a tape that has been overwritten with a smaller amount of data (Option O). BLP is required to position the tape to the beginning of the overwritten data so, once again, the submitted job must be authorized to use BLP. A label number must be specified, pointing immediately after the overwriting data; since a single labeled dataset uses 4 physical files, LABEL=5 is usually appropriate.

```
----- FATAR OVER-WRITTEN TAPE RECOVERY -----
COMMAND ==>                               Enter Q, QI or QO for TMS Query
TAPEIN DD Dsname= BAB.FATARIN.JOBT          <
          Unit  =( TAPE                      )
          Label =( _____,BLP,EXPDT=98000 )
          Volser=( 99999T                      )
          Dcb   =( _____ )
TAPEOUT DD Dsname= BAB.FATAROUT.JOBT        <
          Unit  =( TAPE                      )
          Label =( _____,BLP,EXPDT=98000 )
          Volser=( _____ )
          Dcb   =( _____ )

NUMFILES - Number of Logical files to be read : 0      (zero = all files)
OUTBLP   - By-pass label processing on Output  : YES    (yes/no)
VERIFY   - Verify readability of Output tape  : YES    (yes/no)
EODRETRY - Nbr of attempts to position past EOD: 0
COMPACTION- Data compression request for Output : SYSTEM (system/copy/yes/no)
Display more FATS Operands?                   : NO     (yes/no)
```

Option M (FATAR extended functions) on the FATS/FATAR Main Menu brings up the panel below, which is a starting menu for an *interactive dialog* with which more complex FATAR jobstreams can be built using FATAR's powerful modification/scan control statements.

Option 1 can be invoked one time to specify the input and optional output tapes and generate the JCL. The other options, 2-6, can be invoked repeatedly to generate whatever set of FATAR functions are required in that FATAR step.

```
----- FATAR EXTENDED FUNCTIONS -----
OPTION ==>
1. - ANALYZE data on tape
2. - DROP (Delete) selected blocks or records
3. - KEEP selected blocks or change block length
4. - PRINT selected blocks
5. - REPLACE (Search for) data in selected blocks or records
6. - SCAN data fields in selected blocks or records
7. - RENAME file(s) copied to the output tape
```

Summary

The ISPF panels have been designed to simplify the use of FATS/FATAR for common tasks. Extensive validation of the input is done to eliminate errors at execution time and more complicated tasks, such as dropping blocks or replacing data on a tape, can be much easier to perform.



Corporate Headquarters

Innovation Plaza
275 Paterson Avenue, Little Falls, New Jersey 07424-1658
Tel: (973) 890-7300 Fax: (973) 890-7147
support@fdrinnovation.com sales@fdrinnovation.com
www.innovationdp.fdr.com

European Offices

FRANCE

191, avenue Aristide Briand
94230 Cachan

Tel: (33) 1 49 69 94 02
Fax: (33) 1 49 69 90 98
frsales@fdrinnovation.com
frsupport@fdrinnovation.com

GERMANY

Orleansstraße 4a
D-81669 München

Tel: 089-489 0210
Fax: 089-489 1355
desales@fdrinnovation.com
desupport@fdrinnovation.com

NETHERLANDS (*& Nordic Countries*)

Brouwerstraat 8
1315 BP Almere

Tel: 036-534 1660
Fax: 036-533 7308
nlsales@fdrinnovation.com
nlsupport@fdrinnovation.com

UNITED KINGDOM

Clarendon House
125 Shenley Road
Borehamwood, Herts
WD6 1AG

Tel: 0208-905 1266
Fax: 0208 905 1428
uksales@fdrinnovation.com
uksupport@fdrinnovation.com

C O N C E P T S & F A C I L I T I E S G U I D E