FYI INNOVATION

FOCUS ON:

**FATS/FATAR/FATSCOPY
New Features, User Testimonials
Question & Answers, Audit Reports
and FATSCOPY Resources**

FATS/FATAR, FATSCOPY Version 4.9L28 Now Available!

Enhancements in this new release include:

- IMAGE copy for TLMS is now supported. In version 4.9.27, IMAGE copy for RMM and CA 1 was added. IMAGE copy allows you to make an exact block-for-block copy of the data from the input volume to the output volume, including copying the VOLSER from the input to the output. For data files that have their VOLSER locations recorded in an external data base (such as CA Views and OAM backup tape files), IMAGE allows you to migrate these files to a new tape device without incurring the significant processing overhead of updating the external data base.

- FATSCOPY can now write audit records to a system logger log stream. This allows jobs that are running at the same time to write to a single audit data set, instead of each job writing to a separate sequential data set. This simplifies managing your audit data and generating reports to satisfy auditing requirements. The FATAUDIT utility can produce audit reports from either sequential data sets or from a log stream, and write out data from the logger to a sequential data set for long-term storage.

- FDREPORT, a companion product from INNOVATION, is now available for a 90-day period at no charge to FATSCOPY customers. FDREPORT is a tool that can be used to analyze the tapes in your RMM or CA 1 tape environment to see what resources are in use; determine what types of data sets you have; filter results by creation date, expiration date, and other criteria; and help plan an effective migration strategy.

FATAR USER EXPERIENCES

FATAR continues to benefit customers who find that their physical tape or VTS/DLm data has been corrupted and want to recover as much of the remaining data as possible.

Many data centers are moving away from using real (cartridge) tapes and have migrated to virtual tape systems. While FATAR has traditionally been used for analyzing and correcting errors found on real tapes, FATAR has been used recently to successfully recover data in virtual tape environments.

In November 2013, a customer was receiving ABENDS when using the IBM Sub-Capacity Reporting Tool (SCRT) with one of their SMF log files.

Using FATAR, the customer could see that the SCRT (Sub-Capacity Reporting Tool) abend was caused by an improper spanned logical record, where a record that should have spanned from one block to another was missing one segment of the record. A second FATAR job was used to drop the single record that had incomplete segments. After the file was copied, the SCRT was able to read the file.

**Thanks for all the help!** It was very important to have a full month's worth of SMF data available to feed into the IBM Sub-Capacity Reporting Tool. Without that data, we could have ended up paying Thousands of dollars more for November's 2013 IBM software monthly maintenance charges. I submitted the SCRT report tonight and the report shows "% Data Collected for z/OS 100%" instead of "% Data Collected for z/OS 88%". Anything under 95% is unacceptable.

– A Large University in the MIDWEST

In December 2013, a company found that several of their encrypted 9840D tapes had been improperly re-used, resulting in some of the files on those tapes being overwritten. Using FATAR, the customer was able to identify which files remained on the tape (beyond the overwriting files and the EOD mark), and to copy those files to new tapes. Since FATAR was able to read the user labels (containing the encryption information) from the input tape and copy them to the output tape, the customer was able to successfully read the recovered data sets.

– A Large U.S. Telecommunications Company

FATAR Q & A

Many clients are moving to VTS and they are asking the following:

Q **A year ago, we migrated to a virtual tape system (VTS). Can you explain the benefits or needs for FATS or FATAR with a VTS?**

A FATAR functions on a Virtual Tape System:

1. Investigating a volume, to discover its label type, file count, DCB characteristics, etc., and providing a compact summary of the characteristics of all files on the volume.

2. Examining the data on a volume.

3. Verifying that certain data fields contain valid data.

4. Creating a backup copy of any volume (or multi-volume set of tapes), even if multiple files exist on the volume. FATAR can be used to make a backup physical copy of a virtual tape volume.

5. Verifying that a data set is properly formatted (every block is checked against its DCB info).

6. Correcting invalid data by creating a copy of the input volume(s) with the bad data corrected or dropped.

7. Creating a copy of a volume that was not properly closed (such as during a system failure).

8. Replacing volumes with small blocksizes with copies using a larger blocksize (reducing elapsed and CPU times of applications that read that data).

9. Making an "image copy" (an exact bit-for-bit copy) of a tape volume.

10. Detecting and correcting invalid variable spanned records. An example of this procedure can be found on the Innovation FTP site, http://www.fdr.com/ftp/ftp.cfm, in the file /Public/Download/FATS_FATAR_V49/ FATAR example - correct invalid variable spanned records.pdf

USER EXPERIENCES: See the two recent user experiences described on page 1 on how FATAR users see the benefits of using FATAR for modifying corrupted tape files.

FATSCOPY RESOURCES

Download the FATSCOPY User Manual:
http://www.fdr.com/Manuals_CurrentVersion/FATS_FATAR_FATSCOPY_V49L28_December_5_2013.pdf

How to build a FATSCOPY JCL view the link below:
http://www.fdr.com/Manuals_CurrentVersion/FATSCOPY_How-To-Guide_V49L28_December_5_2013.pdf

Watch the FATSCOPY Product Demo:
http://www.fdr.com/portfolio/products/demo.cfm

Download the FATSCOPY Quick Start guide:
http://www.fdr.com/Manuals_CurrentVersion/FATSCOPY_Quick_Start_Guide_V49L28_December_5_2013.pdf

View the FATSCOPY Concepts & Facilities Guide:
http://www.fdr.com/concepts/FATSCOPYCFG/index.html

The jobstreams and output for examples shown in this newsletter are available from the Innovation FTP site.
- *Go to: www.fdr.com*
- *Click on "FTP login" and Enter your access code (if you don't have one, you can request one on this webpage).*
- *Click on the "Download" directory, then "Tech_Support_Samples/FATSCOPY_Examples/Feb2014".*
- *For FDREPORT examples go to "Tech_Support_Samples/FDREPORT_FATSCOPY_Examples".*

FATAR
Q & A

FATAR examples: identifying overwritten tapes

Q **I accidentally used IEHINITT to re-initialize several of my tapes, making the files on those tapes inaccessible. If I suspect that certain tape volumes were reinitialized, how can I check them to see if they were, so that I can pull them out of the system for later recovery?**

A You can use FATAR to search your tapes for the signature of an IEHINITT job. IEHINITT wipes out the data in the first HDR1 label, replacing it with zeros. You can use the following job to find out if a tape has been reinitialized with IEHINITT:

```
/STEP1       EXEC PGM=FATAR,REGION=0M
...
//SYSIN      DD  *
     ANALYZE LABELS=NO,NF=1
     REPLACE LF=1,B=ALL,S=C'HDR10000000000000000000000000000000000'
```

*FTP member: **FATAR_IEHINITT_jcl.txt*** ← [Sample JCL]

LABELS=NO tells FATAR to act as if the tape is unlabeled, so FATAR will scan the label as if it's ordinary data. If the HDR1 label contains a string of zeros, a BLOCK MODIFIED message will appear in the SYSPRINT output. (No blocks are actually modified since there is no TAPEOUT file; the input tape is not changed.)

Here is a sample output:

```
                                    IEHINITT Job

* * * * * * * START FILE       1
      1     80  INPUT LABEL              VOL1STKE09
              ***VOLUME LABEL***              VOLSER=STKE09 OWNER=
      2     80  INPUT LABEL              HDR1000000000000000000000000000000000000000000000000000000000000000000000000000
              ***HEADER LABEL***              DSNAME=00000000000000000 SERIAL=000000 VOLSEQ#=0000 FILESEQ#=00000
                                              CRTDT=000000 EXPDT=000000 SECURITY=0 BLOCKCOUNT=0000000000 BLKID=000A0000
              BLOCK MODIFIED
* * * * * * * END OF FILE      1  -- FILE CONTAINED        2 BLOCKS
FATS022 ANALYSIS TERMINATED AFTER SPECIFIED NUMBER OF FILES
```

*FTP member: **FATAR_IEHINITT_output.txt*** ← [Sample Output]

Q **I relabeled 100 tapes. One of those tapes was not a scratch tape and should not have been relabeled. How can I find which of these tapes was the volume I want to keep so that I can pull it out of the system for later recovery?**

A You can use FATAR to search each tape for an identifying string which you expect to find on the original volume. This string will be located somewhere after the overwriting label. The original volser will be found in the file labels and you can scan for that. For example, to look for volume STKE10, use:

```
/STEP1       EXEC PGM=FATAR,REGION=0M
...
//SYSIN      DD  *
     ANALYZE LABELS=EOD,LBLPRT=FORMAT
     REPLACE LF=ALL,B=ALL,S=C'STKE10'
```

*FTP member: **FATAR_find_vol_jcl.txt*** ← [Sample JCL]

LABELS=EOD tells FATAR to skip past the EOD mark (written when you relabeled the tape) before searching for the residual data remaining on the tape. The file HDR and EOF labels will be displayed in the SYSPRINT output with a BLOCK MODIFIED message if the string 'STKE10' is found. (No blocks are actually modified since there is no TAPEOUT file; the input tape is not changed.)

*FTP member: **FATAR_find_vol_output.txt*** ← [Sample Output]

FATSCOPY AUDIT

Q **I am setting up a complex migration project. How can I keep a record of which data sets and volumes have been copied without having to save and sort through hundreds of job logs?**

A FATSCOPY can write records to an audit data set which contains information on the data sets copied (the input and output volumes, expiration dates, etc.) by that job. A second copy job can append to that audit data set, and then a third, and so forth. You can run the FATAUDIT utility at any time which reads the audit data set and formats out a report displaying the files copied and summary information for all of the jobs which were recorded in that audit data set.

Here's an example of a job which writes to a new audit data set:

*FTP member: **Copy_With_New_Audit.txt***

```
//JOBLIB      DD   DSN=FATCOPY.LIBRARY,DISP=SHR
//AUDIT1      EXEC PGM=FATSCOPY,REGION=0M
...
//AUDIT       DD   DSN=JAT.FATSCOPY.AUDITDSN,UNIT=SYSALLDA,
//                 DISP=(NEW,CATLG),SPACE=(TRK,(5,5))
//TAPEOUT     DD   DSN=DUMMY,DISP=(NEW,KEEP),UNIT=VTAPE
//SYSIN       DD   *
  COPY
  SELECT ALLDSN,VOL=A00001
  SELECT ALLDSN,VOL=B34220
/*
```

A second job which appends to that audit data set would change the AUDIT DD to:

```
//AUDIT       DD DSN=JAT.FATSCOPY.AUDITDSN,DISP=MOD
```

but otherwise looks the same as the first job, with new SELECT statements.

Here is a FATAUDIT job which reads the audit data set to produce a printable report:

*FTP member: **Write_Audit_Report.txt***

```
//AUDIT2      EXEC PGM=FATAUDIT,REGION=0M
//SYSABEND    DD   SYSOUT=*
//AUDITRPT    DD   SYSOUT=*
//AUDIT       DD   DSN=JAT.FATSCOPY.AUDITDSN,DISP=SHR
/*
```

FATSCOPY AUDIT

Here is a sample of a portion of an Audit Report produced by FATAUDIT:

```
                              FATAUDIT DETAIL REPORT

            JOBNAME=JATPAN      STEPNAME=FATSCOPY   RUN DATE= 2013303   RUN TIME=  145653    CPUID=CPUC


                              ---------  INPUT INFORMATION  ------------¦-------  OUTPUT INFORMATION   ----
 DATA SET NAME                 VOLSER  FSEQ  VSQ  M/V EXPDATE CAT SCRDATE VOLSER  FSEQ  VSQ EXPDATE CAT TMS RC      REAS
 NEW NAME                         BLOCKS            BYTES                    BLOCKS            BYTES
 -----------------------------------------------------------------------------------------------------------------------
 JAT.EE.FATS.FILE1             CCR009   1    1   NO  2019001 YES          E10037   1     1 2013304 NO   YES 0000
                                   2              0                           2              0
 JAT.EE.FATS.FILE2             CCR009   2    1   NO  2019002 YES          E10037   2     1 2013304 NO   YES 0000
                                   2              0                           2              0
...
                              FATAUDIT SUMMARY REPORT
   TOTAL INPUT VOLUMES   -          25
   TOTAL OUTPUT VOLUMES  -          25
   TOTAL BYTES READ      -  9550256056
   TOTAL BYTES WRITTEN   -  9550256056
   TOTAL BLOCKS READ     -     1463991
   TOTAL BLOCKS WRITTEN  -     1463991
   TOTAL FILES COPIED    -         486
   TOTAL FAILED COPIES   -           0
   TOTAL JOBS EXECUTED   -          46
```

Q **OK, but I want to run two (or more) FATSCOPY jobs at a time. Can I write to the same audit data set with copy jobs that are running concurrently?**

A You can't use the same sequential audit data set for two jobs which run at the same time. There are two options you can use: use separate audit data sets and merge them into one Audit Report, or use a System Logger log stream to write audit records to a single log stream.

- One "set" of copy jobs can use an audit data set MY.AUDIT.DATASET1 and a second "set" can use MY.AUDIT.DATASET2. When you want to produce an Audit Report, the FATAUDIT utility can use both audit data sets as input and produce a single Audit Report.
  Here's how you use 2 audit data sets to produce 1 report:

*FTP member: **Audit_Report_Using_Two_Audit_Datasets.txt***

```
//AUDIT      EXEC  PGM=FATAUDIT,REGION=0M
//AUDITRPT   DD    SYSOUT=*
//AUDIT      DD    DSN=MY.AUDIT.DATASET1,DISP=SHR
//           DD    DSN=MY.AUDIT.DATASET2,DISP=SHR
/*
```

- Instead of writing audit records to a sequential data set, FATSCOPY can write audit records to a system logger log stream. The z/OS system logger is a component of z/OS that must be activated and configured. If you have configured a log stream to receive FATSCOPY audit records, then your copy jobs will look like:

*FTP member: **Copy_Using_System_Logger.txt***

```
//JOBLIB     DD    DSN=FATCOPY.LIBRARY,DISP=SHR
//AUDIT1     EXEC  PGM=FATSCOPY,REGION=0M
...
//TAPEOUT    DD    DSN=DUMMY,DISP=(NEW,KEEP),UNIT=VTAPE
//SYSIN      DD    *
   COPY AUDITLOG=MY.FATSCOPY.LOGSTRM
   SELECT ALLDSN,VOL=A00001
   SELECT ALLDSN,VOL=B34220
...
```

and audit records are written to the log stream. Concurrent jobs can write to the same log stream. The FATAUDIT utility can read audit records from the log stream and produce Audit Reports.

FDREPORT

BASIC REPORTS

FDREPORT is a planning tool which can be used with RMM or CA 1 to query the tape management database using a broad range of powerful selection parameters.

FDREPORT is available to FATSCOPY customers at no charge for 90 days.

FDREPORT generates reports on the tape environment that you can use to determine which volumes you need to copy.

Sample FDREPORT volume usage reports FOR RMM AND CA1

RMM REPORT - LIST OF VOLUMES & DATA SETS NOT EXPIRED IN VOLSER ORDER

VOLSER	DATA SET NAME	EXPDAT	EXDAY	%FU	APPROXSIZE	FILE SIZE	PHYSICALSIZ
200154	A03.A04345.RSDGR07.C109244.T023014.E2014338	2014.345	374	3	12619.874M	12619.612M	643825.664K
200159	A03.A04345.RSDBK07.C109244.T023015.E2014338	2014.345	374	3	12619.874M	12619.612M	13631.488K
200161	ANICO1.SMF.ESA.JAN12	2014.062	91	7	23410.199M	23409.459M	5249.171M
…							

RMM REPORT - VOLUMES NOT IN SCRATCH STATUS:

VOLSER	VOLSTAT	CRDATE	EXPDAT	RTDATE	RETENTVALU	DSCOUNT	PREVOL	NXTVOL	OWNER
FAT49P	MASTER	2008.088	1999.365	2008.093		1			SS001
FDR54P	MASTER	2008.088	1999.365			0			SS001
I00000	MASTER	2003.297	2012.358	2014.022		29	I03745	I00094	SS040
…									

FOR CA1 - LIST OF VOLUMES & DATA SETS NOT EXPIRED & NOT EDM, IN VOLSER ORDER

VOLSER	VOLSTAT	NDSNB	%FU	CRDATE	CRDAY	EXPDAT	EXDAY	CA-1 DENSITY
800034		1	0	2013.119	205	2014.119	160	3590 CART TAPE
800041		0	7	2010.341	1079	1999.000	65535	3590 CART TAPE
800044		0	1	2012.286	404	2016.102	873	3590 CART TAPE
…								

Sample FDREPORT Summary Reports

```
FDR400   FDRABR REPORT STANDARD SUMMARIES - FDREPORT VER 5.4/80   - INNOVATION DATA PR..
RMM REPORT -   TOTAL VOLUMES NOT IN SCRATCH STATUS:
     FINAL TOTALS --
     TVVOLSER-----  200287  ◄─────── TOTAL VOLUMES NOT IN SCRATCH STATUS
```

```
FDR400   FDRABR REPORT STANDARD SUMMARIES - FDREPORT VER 5.4/80   - INNOVATION DATA PR..
RMM REPORT -   TOTAL VOLUMES IN SCRATCH STATUS:
     FINAL TOTALS --
     TVVOLSER-----  61411  ◄─────── TOTAL VOLUMES IN SCRATCH STATUS
```

```
FDR400   FDRABR REPORT STANDARD SUMMARIES - FDREPORT VER 5.4/80   - INNOVATION DATA PR..
RMM REPORT - APPROX SIZE OF ALL EXPIRED DATA SETS & (NOT SCRATCH)
     FINAL TOTALS --
     TDAPRSIZ---  59.704T  TDVOLSER------138767
```
APPROX SIZE OF ALL EXPIRED DATA SETS & (NOT SCRATCH) Total Terabytes

```
FDR400   FDRABR REPORT STANDARD SUMMARIES - FDREPORT VER 5.4/80   - INNOVATION DATA PR..
RMM REPORT - APPROX SIZE OF ALL DATA SETS NOT EXPIRED & (NOT SCRATCH)
     FINAL TOTALS --
     TDAPRSIZ---  102.286T  TDVOLSER-------61569
```
APPROX SIZE OF ALL DATA SETS NOT EXPIRED & (NOT SCRATCH) Total Terabytes

Here we show a simple FDREPORT example of GROUPING on the VOLUME STATUS (MASTER,SCRATCH,…) Grouping can be used with many other fields such as expiration date, creating program/job, last reference date, and more. With groupings you can get a good overview in (sub)totals per grouping (MASTER/SCRATCH) and the final totals.

```
         LIST ALL TAPES GROUPED PER VOLUME STATUS SCRATCH - EXTRACT FILE 2013.329

  VOLSER VOLSTAT    EXPDAT   LRDATE LOCATION
  ------ -------  -------- -------- --------
  100029 SCRATCH           2009.190 AWAATL1
  100060 SCRATCH           2011.274 AWAATL1
  .......
  104610 SCRATCH                    AWAATL1
      SUBTOTAL -- TVSTATUS--SCRATCH
          TVVOLSER--------94
```

```
         LIST ALL TAPES GROUPED PER VOLUME STATUS MASTER - EXTRACT FILE 2013.329

  VOLSER VOLSTAT    EXPDAT   LRDATE LOCATION
  ------ -------  -------- -------- --------
  100048 MASTER   2010.062 2010.057 AWAATL1
  100049 MASTER   2007.310 2013.239 AWAATL1
  100050 MASTER   2009.295 2009.290 DISTANT
  .......
      SUBTOTAL -- TVSTATUS--MASTER
          TVVOLSER--------4518
      FINAL TOTALS --
          TVVOLSER--------  4612   ◄──── TOTAL FOR BOTH SCRATCH AND MASTER
```

Tape Volume Usage Report

```
                 VOLUME USAGE REPORT  - EXTRACT FILE 2013.326

  VOLSER VOLUSEBYTES %FU
  ------ ----------- ---
  200002 21041.650M  29
  200003  3694.789M   0
  200004 21041.650M  29
  200005 28800.102M  32
  200007  3694.789M   0
  200011     0.000M   7
  200015 22052.438M  31
  200017 20815.544M   7
  200021  5612.859M   1
  200022 46852.009M  26
  200024     0.000M   3
  200025  3694.789M   0
  ........
  253782     7.082M   1
  253783     0.029M   0  ◄──── TOTAL VOLUMES
  253784   210.022M  12
  253785   197.869M   8
    FINAL TOTALS --
        TVVOLSER----- 5203   TVUSEBYT-- 38548.046G  ◄──── TOTAL GIGABYTES
```

You should use this report to identify tapes that have a history of errors.
You might want to process these tapes separately.

```
            TAPES WITH PERMANENT ERRORS - EXTRACT FILE 2013.326

  LUNI VOLSER  LWRDAT   LRDATE  PERMRDERR PERMWTERR
  ---- ------ -------- -------- --------- ---------
  0801 201638 2006.251 2006.251        92         2
  080E 200204 2010.253 2013.254       116         0
  080F 201450 2010.252 2010.271       106         0
  0810 202032 2010.212 2010.212       211         0
  080F 202740 2011.358 2012.141        38         0
```

NOTE: For more information on how to use FDREPORT with FATSCOPY view:
http://www.fdr.com/Using FDREPORT for Tape Migration with FATSCOPY.pdf
See page 2 on how to access these sample FDREPORT jobs.

FATSCOPY Checkpoint/Restart examples:

Q I have long-running copy jobs that I sometimes have to interrupt at the end of my processing window. How can I make it easy to pick up where I left off?

A FATSCOPY's checkpoint/restart features allow you to stop a running job with an operator command (STOP or CANCEL) and save the status of the current job in a DSNTABLE data set. The DSNTABLE data set saves a list of the files which were selected for the copy job, and marks which of those files have been copied. A FATSCOPY RESTART job using that DSNTABLE data set loads the list of files, skips those which have already been copied, and copies the selected files starting with the first non-copied file.

Here's an example of a job which might select a large number of data sets:

FTP member: Large_Copy_Job.txt

```
//JOB1       EXEC PGM=FATSCOPY,REGION=0M
//STEPLIB    DD   DSN=FATSCOPY.LIBRARY,DISP=SHR
...
//DSNTABLE   DD   DSN=JAT.FATSCOPY.RESTART,UNIT=SYSALLDA,
//                SPACE=(TRK,(15,5),RLSE),DISP=(,CATLG)
//TAPEOUT    DD   DSN=DUMMY,UNIT=VTAPE,DISP=(,KEEP)
//SYSIN      DD   *
 COPY MULTIFILE=YES
 SELECT   CATDSN=PROD.**
/*
```

The operator stops the job with a STOP command while the 259th selected file is being copied. FATSCOPY finishes copying the 259th file and then terminates, writing out the data set JAT.FATSCOPY.RESTART. The following job is used to continue copying starting with the 260th file:

FTP member: Restart_Job.txt

```
//JOB2       EXEC PGM=FATSCOPY,REGION=0M
//STEPLIB    DD   DSN=FATSCOPY.LIBRARY,DISP=SHR
...
//DSNTABLE   DD   DSN=JAT.FATSCOPY.RESTART,DISP=SHR
//TAPEOUT    DD   DSN=DUMMY,UNIT=VTAPE,DISP=(,KEEP)
//SYSIN      DD   *
 COPY RESTART
/*
```

FATSCOPY Q & A

Q **I want to combine as many files onto a single tape or set of tapes, but I want to do this in separate jobs run over the course of a week. How can I do this without having to manually keep track of where the output files have been written?**

A You can use FATSCOPY's LASTAPE feature. When you use LASTAPE for the first time in a copy job, such as in this example:

*FTP member: **Copy_Using_LASTAPE.txt***

```
//FATSCOPY EXEC PGM=FATSCOPY,REGION=0M
//STEPLIB   DD   DSN=FATSCOPY.LIBRARY,DISP=SHR
...
//TAPEOUT   DD   DSN=DUMMY,UNIT=VTAPE,LABEL=(,SL)
//SYSIN     DD   *
  COPY EXPDTGROUP=999,LASTAPE=MY.LASTAPE.DATASET
  SELECT ALLDSN,VOL=CCR009
  SELECT ALLDSN,VOL=001122,NUMVOLS=10
...
/*
```

EXPDTGROUP=999 tells FATSCOPY to stack as many files onto as few tapes as possible. FATSCOPY catalogs a dummy file (with a name you choose as the LASTAPE= parameter) at the end of the last output tape in that copy job. The next time you run a copy job using the same LASTAPE file name, FATSCOPY determines that the LASTAPE file already exists, finds out which volume it is on, and continues to write files on that volume. At the end of the job, it recatalogs the dummy file at the (new) end of the output tape. You can repeat this to keep adding files to the volume you have been writing to.

Q **I have ranges of input volsers I want to copy to a single high-capacity output tape. How can I do this?**

A The FATSCOPY keywords NUMVOLS (to specify a range of consecutive volsers) and EXPTDGROUP=999 (to stack all data sets onto as few volumes as possible) make this easy to do:

*FTP member: **Stacking_Volume_Ranges.txt***

```
//FATSETUP   EXEC   PGM=FATSCOPY,REGION=0M
//STEPLIB    DD     DSN=FATSCOPY.LIBRARY,DISP=SHR
...
//TAPEOUT    DD     DSN=DUMMY,UNIT=MYTAPE,DISP=(,KEEP)
//SYSIN      DD     *
 COPY EXPDTGROUP=999
 SELECT ALLDSN,VOL=003200,NUMVOLS=100   ◄── This will select volser 003200 to 003299
 SELECT ALLDSN,VOL=003400,NUMVOLS=50
/*                                          ◄── This will select volser 003400 to 003449
```

FATSCOPY

Q & A

Q **I have a tape-backed TS7740 virtual tape device. I want to make backup copies of the logical volumes in the VTS while minimizing the number of physical tape mounts. What is the best way to do this with FATSCOPY?**

A FATSCOPY has two keywords, PHYSVOL and ALLPHYS, which allow you to select and copy all of the logical volumes from a single physical volume in a single job. If you know the physical volser you want to copy from, use PHYSVOL. If you know the logical volser of one logical volume on the physical tape, use ALLPHYS. Here's an example for when you have a logical volser and you want to find and copy all the logical volumes on the same physical back-end tape:

FTP member: Copy_Using_IBM_PHYSVOL.txt

```
//STEP1     EXEC PGM=FATSCOPY,REGION=0M
//STEPLIB   DD  DSN=FATSCOPY.LIBRARY,DISP=SHR
...
//TAPEOUT   DD  UNIT=OUTDEV,DSN=DUMMY,DISP=(,KEEP)
//MAPTAPE   DD  UNIT=(INPUTDEV,,DEFER),DISP=(,CATLG),
//              DSN=MY.MAPTAPE.FILE,LABEL=(,SL),
//              DCB=(RECFM=F,LRECL=80,BLKSIZE=80,TRTCH=NOCOMP)
//SYSIN     DD  *
 COPY  VIRTTYPE=IBM
 SELECT ALLDSN,ALLPHYS=CCR012
/*
```

The MAPTAPE data set is used to communicate tape mapping information between the VTS and FATSCOPY. FATSCOPY will find out which physical volume that logical volume CCR012 is located on, and select and copy all the logical volumes on that physical volume.

NOTE: a similar function is available for Oracle StorageTek tape-backed virtual tape devices.

PRODUCT UPDATES

This table shows, for each current release of z/OS, the minimum levels of each Innovation product that should be installed. Use the table when upgrading your z/OS system, to verify that you have the correct Innovation software in use. If not, you should install the latest version. If you already are using the indicated minimum level, you may wish to install the latest version anyway in conjunction with the Operating System upgrade. If not, please review the Product Notes that follow the table on our website at www.fdr.com/osreq.cfm for ZAPs that may be required depending on the hardware and software in use at your installation.

MAINFRAME SERVER SUPPORT
zEC12 Mainframe Server Support requires no changes to the supported Innovation Products.

As of February 2014

PRODUCT	z/OS 1.10 -1.12 Minimum	z/OS 1.13 Minimum	z/OS 2.1 Minimum	Latest Version
FDR	V5.4/74	V5.4/74	V5.4/78	V5.4/80
COMPAKTOR	V5.4/74	V5.4/74	V5.4/78	V5.4/80
ABR	V5.4/74	V5.4/76	V5.4/78	V5.4/80
FDRCRYPT	V5.4/74	V5.4/74	V5.4/78	V5.4/80
FDREPORT	V5.4/74	V5.4/74	V5.4/78	V5.4/80
FDRViEWS	V5.4/74	V5.4/74	V5.4/78	V5.4/80
FDRINSTANT	V5.4/74	V5.4/74	V5.4/78	V5.4/80
FDRDRP	V5.4/74	V5.4/74	V5.4/78	V5.4/80
FDRREORG	V5.4/74	V5.4/74	V5.4/78	V5.4/80
FDRPAS	V5.4/76	V5.4/76	V5.4/78	V5.4/80
FDRPASVM	V5.4/80	V5.4/80	V5.4/80	V5.4/80
FDRMOVE with FDRPAS	V5.4/76	V5.4/76	V5.4/78	V5.4/80
FDRERASE for z/OS	V5.4/70	V5.4/70	V5.4/70	V5.4/70
FDRSOS	V5.4/76	V5.4/76	V5.4/78	V5.4/80
FATS/FATAR/FATSCOPY	V4.9/20	V4.9/20	V4.9/25	V4.9/28
IAM	V8.1/18	V8.1/18	V9.0/04	V9.1/01
UPSTREAM CLIENT	V3.6.0	V3.6.0	V3.6.0	V3.7.28
UPSTREAM for System z	V3.6.0	V3.6.0	V3.6.0	V3.7.28
UPSTREAM z/OS UNIX	V3.5.0	V3.5.0	V3.6.0	V3.7.1
UPSTREAM Director	-	-	-	V4.2.01
UPSTREAM z/OS Storage	V3.6.0	V3.6.1	V3.7.0	V3.8.0
UPSTREAM Reservoir	-	-	-	V3.7.2e

For the most up-to-date information visit the website at: http://www.fdr.com/osreq.cfm

NOTE: While prior versions might still work, the versions listed are the minimum supported versions.

VISIT US AT:
SHARE Technology Exchange: March 10-12, 2014   Anaheim CA
EMC World 2014: May 5-8, 2014   Las Vegas, NV
SHARE Technology Exchange: August 4-6, 2014   Pittsburgh, PA

FDREPORT

WITH FATSCOPY

FDREPORT: A Companion Tool Available with FATSCOPY

FDREPORT is a planning tool which can be used with RMM or CA 1 to query the tape management data base using a broad range of powerful selection parameters.

FDREPORT generates reports on the tape environment that you can use to determine which volumes you need to copy. Sample selection criteria:

- Data sets expiring after a certain date
- Data sets created within a specified date range
- Data sets created by specified programs
- …etc.

For an EMC DLm, FDREPORT can generate scratch volume reports that can be used as input to the DLm utility for scratching the backing disk space of logical tape volumes that are in scratch status.

FDREPORT is available to FATSCOPY customers at no charge for 90 days.

FDREPORT helps to properly plan a customer's migration.

FDREPORT can report on:

- How many tapes are
  - Active
  - In scratch status
  - Managed by an EDM
- % of space used on each tape
- Number of tapes and capacity expiring in nnn days

You can generate these reports from the customer's active RMM/CA 1 data base.

You can also create an extract file from the tape management data base, **allowing you to generate reports on your own system** instead of at the customer's site. This extract file is very helpful for those 3rd parties/VARs that are contracted to do a tape migration.

Sample FDREPORT Summary Reports

```
FDR400   FDRABR REPORT STANDARD SUMMARIES - FDREPORT VER 5.4/80   - INNOVATION DATA PR..
RMM REPORT -   TOTAL VOLUMES NOT IN SCRATCH STATUS:
    FINAL TOTALS --
       TVVOLSER-----  200287      ◄────   TOTAL VOLUMES NOT IN SCRATCH STATUS
```

```
FDR400   FDRABR REPORT STANDARD SUMMARIES - FDREPORT VER 5.4/80   - INNOVATION DATA PR..
RMM REPORT - APPROX SIZE OF ALL DATA SETS NOT EXPIRED & (NOT SCRATCH)
    FINAL TOTALS --
       TDAPRSIZ---  102.286T    TDVOLSER-------    61569     ◄────   TOTAL VOLSER
                        │
                        └──── TOTAL TERABYTES
```

See pages 6 and 7 for more details on how to use FDREPORT with FATSCOPY.
A pdf of this newsletter is available at www.fdr.com/FYINewsFATSCOPY.pdf