T R I C K S W I T H T A P E S

Sam Golob sbgolob@cbttape.org

August 19th, 2004

Session 2824

S H A R E

Session 2824

T R I C K S W I T H T A P E S

Speaker: Sam Golob (SGO)

Tape Basic Concepts

- Where are “tapes” created?
  - Tape drive
  - Virtual tape as a disk file
  - Virtual tape in a VTS
- What will be covered today
  - General Information
  - Tools
- Books
  - Hardware
  - Software
    - Using Magnetic Tapes
    - DFDSMSdfp: Advanced Services

Tapes are a "simple" type of I/O, because they are one-dimensional.

Tapes are created in:

1. A tape drive
2. Virtually, as disk files
3. In a virtual tape library

Today I'm going to divide this talk into two parts:

1. General info about the types of processes involved that have to do with tapes.
2. Some specific tools (almost all free) that will allow you to do what you want with tapes.

There are three types of books that you'll need.

1. The hardware book(s): 3480, 3490, 3490E, 3590
   For op codes in coding channel programs.
2. Software: Using Magnetic Tapes
   For Standard Label processing information
   For ASCII Label processing information
3. DFDSSdfp: Advanced Services
   For information on how to use EXCP.

   3A. It would also be useful to see the Data Management Macros book for the BSAM and QSAM macros, if you want to use them.

IBM Access Methods (for tape)

- Standard IBM Access Methods
  - EXCP
  - BSAM
  - QSAM
- Languages
  - Assembler
  - Everything else
- BLP
  - APF

Standard IBM Access Methods that are used to read and write to tapes:

1. EXCP
2. BSAM
3. QSAM

WE LIKE EXCP !!! (But BSAM is tolerable.)

Languages are optional:

But we like ASSEMBLER !!!

One of the tricks we do with tapes, is to treat Standard Labels (if they exist) as plain files. In order to do this, you need:

1. BLP privilege either in the JES initiator, or in RACF.

2. APF Authorization, to force BLP privilege in the program.

The reason why you need BLP privilege is to cause the MVS system to ignore the contents of the tape labels.

BTW, DASD (disk) packs also have standard labels.
The system uses tape labels in a similar manner to how it uses disk labels.

Data on tapes

- Labels
- Files
- Tape Marks

Data on tapes:

Labels, Files, and Tape Marks

(On 3590s there's some kind of "end of tape" marker.)

Files consist of (a sequence of separate) blocks of data,
separated by a tape mark.

Labels really are files too, but the system uses them.

Tape marks separate files. Two tape marks in a row, usually
end the data on a tape. (But not always--for instance, when
you have a null SL file, you have the HDR labels, then 2 tape
marks, then the EOF or EOV labels.)

Files can be FB, VB, or U, the same way disk data is.

SL files are really one group of three files.

1 HDR label group (VOL1 label first, if first file on the tape)
2 the data blocks
3 EOF or EOV label group

EBCDIC (IBM) Tapes, and ASCII (ANSI/ISO) Tapes

- EBCDIC (IBM) Tapes
  - Standard Labels supply most need information
- ASCII (ANSI/ISO) Tapes
  - MVS does support ASCII tapes
  - Two “LEVELS”

EBCDIC (IBM) Tapes, and ASCII (ANSI/ISO) Tapes.

Most of the tapes we deal with, in an MVS shop, are IBM tapes.

The data on IBM EBCDIC tapes can be anything.

The Standard Labels on EBCDIC tapes are usually all we need to
know about.

But MVS does actually have some support for ASCII tapes (usually
tapes coming from, or going to, other systems).

The "Using Magnetic Tapes" book spends half its space on ASCII tapes.
(This is probably because they are a world unto themselves.)

There are two commonly used "LEVELS" or standards, for ASCII tapes.
The 80th character of the VOL1 label has the level number.

   Level 3, and Level 4 are supported by MVS.
   Level 1 is old, and is supported only for "read only".

Level 4 is much more flexible than Level 3.

   Level 3 has a blocksize limit of 2048.
   Level 3 only supports ASCII data files.
   You can't have two identical file names on a Level 3 ASCII tape.

My limited experience with Level 3 ASCII tapes is that they are
a big pain in the neck. But for compatibility with other systems,
they have to be supported. DFSMSdfp 1.5 and higher, support
ASCII LEVEL 4 tapes. On lower levels, IEHINITT won't even INIT
a LEVEL 4 tape, only LEVEL 3.

d

Copying Tapes

- Don’t use IEBGENER!
  - IEBGENER copies files not tapes
  - IEBGENER wants you to tell it DCB information
  - IEBGENER may reblock the data without telling you
  - IEBGENER provides no tools to find out what is on the tape
  - So… Don’t use IEBGENER
- Tools
  - DITTO
  - RMM
  - VENDOR tools
    - TelTape from www.cartagena.com
- What does the “Ordinary Joe” do?

Copying Tapes.

Early in my career (I was an application programmer), I asked some application programmers: "How do you copy a tape?"

They answered: "Use IEBGENER".

THIS IS A BIG MISTAKE !!!

Reasons:

1 IEBGENER only copies one file at a time.
2 You have to tell IEBGENER DCB attributes. Newer versions of IEBGENER will (sneakily) reblock the data without telling you. You lose file integrity on the copied tape.
3 IEBGENER doesn't give you any tools to help you find out what is on the tape in the first place. How many files, etc.

IBM doesn't really supply any tools at all for copying tapes, except:

1 DITTO (an optional "separate pay" product)
2 migration tools for RMM or automated tape libraries (not available to everyone)

Vendors have tape migration tools:

A good one: TelTape from www.cartagena.com

Vendor suppliers of automated tape libraries had better supply tape copying and migration tools, or they'd be out of business.

BUT WHAT DOES THE "ORDINARY JOE" DO?

Copying Tapes
Tools for an ORDINARY JOE

- Joe is really the “system doctor”, the SYSPROG
  - IBM did not provide Joe tools. Why?
- Tape copy program is “simple”
  - REWIND both tapes
  - Read a block of data
  - If a tape mark was read write a tape mark
  - Write a block of data from the buffer to output
  - If you encounter an end of tape condition write TWO tape marks on output and get out


```
Copying Tapes.  Tools for ORDINARY JOE.

Joe isn't so ordinary.  Joe is the "system doctor", the SYSPROG.

Joe is supposed to be able to do "magic".

So why doesn't IBM give Joe any tools for copying a tape from
end to end?

   (I think)   Beause IBM is worried about security.

               You can take a copied tape offsite easily.

               IBM isn't worried that you'll put a 3390
               disk group in your pocket and walk out
               with it.  Even the old portable 3340s were
               quite bulky.

But copying a tape is very easy in concept:  You just have to:

  1.  Rewind both tapes to the beginning (optional).

  2.  Read a block of data into a buffer in storage.

  3.  If you encounter a tape mark on input, write a tape mark
      on output.

  4.  Write the block of data from the buffer to the output
      tape.

  5.  If you encounter an end of tape condition, write two
      tape marks to the output tape(s) and get out.
```

Original COPYMODS program (by Paul Tokheim) - CBT Tape File 229

- COPYMODS from Paul Tokheim
- Original had limitations
  - 32K blocksize
  - Distinction of labels and data files
- COPYMODS (original and current) did not require APF but you need to have BLP privileges!


```
The original COPYMODS program (from Paul Tokheim) - CBT Tape
File 229:

It did just that.  Except that since the data was in a buffer,
it could be copied to more than one output tape.

It was fine, except that it was a "first cut".  The original
COPYMODS didn't have enough "smarts".

   1.  COPYMODS didn't copy tape blocks bigger than 32K.
       EXCP gives you 64K "for free".

   2.  The old COPYMODS didn't know the difference between
       labels and data files.  Therefore:

       A.  If you had a "null SL" file on a tape, which
           had two consecutive tape marks on it "legally",
           COPYMODS would write the HDR group and stop after
           the two tape marks, leaving the rest of the tape
           files "orphaned" and NOT COPIED.

COPYMODS (original and current) has one advantage.

   You don't have to run it APF Authorized.  You only need to
   have BLP privilege.

   OF COURSE, YOU NEED BLP !!!
```

Tape Mapping and Measuring

- What is on a tape?
- Non-Free Tools
  - DITTO (IBM) which is not free
  - IBM may provide other tools
  - VENDOR Tools like FATS/FATAR
- Free Tools Bonanza!
  - http://www.cbttape.org to get these free tools!
  - 1. TAPEMAP (CBT File 299, and Leonard Woren's)
  - 2. TAPESCAN (CBT File 102)
  - 3. COPYMODS (CBT File 229) - I mean the NEW COPYMODS
  - 4. SS0104 (CBT File 266)
  - 5. TCOPY (CBT File 193)
- Mapping and Copying tapes are related activities

Tape Mapping and Measuring.

All that stuff is well and good when it comes to copying a tape.
But what if you just need to find out what is on a tape, and
extract one or more of the files from it?

You need a "Tape Mapping" program for that.

Does IBM provide you with one? Debatable.

1. DITTO can do it. (But it's an "extra pay" component.)

2. After that, I'm not sure.

Vendors sell tape mapping tools.

One example: FATAR from Innovation.

But here's where the users have written most of the good stuff !!

Go to our www.cbttape.org web site.

Examples:

1. TAPEMAP (CBT File 299, and Leonard Woren's)
2. TAPESCAN (CBT File 102)
3. COPYMODS (CBT File 229) - I mean the NEW COPYMODS
4. SS0104 (CBT File 266)
5. TCOPY (CBT File 193)

Mapping tapes, and Copying tapes, are TWO RELATED ACTIVITIES.

MAPPING Tapes, as related to COPYING Tapes

- MAPPING vs. COPYING
  - Both READ the contents
- What to report in a tape “MAP”?
  - Tape Labels
  - Data – blocksize, number of blocks, footage, byte counts
  - Data printed samples of data
  - Data statistics

MAPPING Tapes, as related to COPYING Tapes.

In both cases, you have to READ the contents of the tape.

As you are reading, you can gather as much information as you want. Remember that tapes are one-dimensional!

If you've already gathered the information, why not report it.

"If you've got it, flaunt it!"

What should you flaunt? Plenty: (I'm still learning....)

1. Label information - contents of the various label fields.

2. Data information - you can scan the data itself to find out

   A. Physical information - blocksizes, number of blocks, footages, byte counts, etc.

   B. Telltale information about which program produced this file. Once you know the program, you often can figure out a lot about the contents too.

   C. HEX and EBCDIC (or ASCII) print, to find out more detail about the tape data AND the labels.

   D. Statistics about the tape contents and file contents.

All this stuff should (at least) have the result that you can then extract any data you want, off the tape.

It shouldn't make much difference if you're copying the tape or merely reading it to map it. The same info is available to the program. That's why some programs DO BOTH TAPE MAPPING AND TAPE COPYING too. TAPES ARE ONE-DIMENSIONAL !!!!!

AWS Tape Format "Tapes"

- IBM invented the AWS format because it was needed on the OS/2 based P/390 mainframe emulator
- AWS is disk file on the emulator visible to the guest operating system as a tape
- Hercules and Flex-ES also provide capabilities
- The AWS format has uses beyond the emulators
- AWS is a portable format for tape useful for archiving or transmission

AWS Tape Format "Tapes"

When IBM invented the P/390 as an OS/2-based machine--essentially a PC--which simulated S/390 machine instructions and I/O, they also invented a way to represent a "physical tape" as a disk file.

In the P/390, the format of the disk file which the P/390 can read as a "tape" is called AWS format. It was quite simple to call it that--AWS is the IBM three-letter prefix for all of the P/390 driving software.

Other vendors have also simulated S/390 instructions and I/O on a PC. Notable is FLEX-ES from Fundamental Software, with uses a UNIX-based simulation. There are others too, such as the Linux-based Hercules system. Each of these systems can read disk-based simulated "tapes".

If you're operating in the modern world, you should be aware of what PC-based "tapes" can offer. For example, you can archive all of your magnetic tapes on CD-rom and DVD-rom. And you don't need a P/390 or a Flex system or Hercules to do it. (See CBT Tape File 533.)

Particulars of the AWS Format for "Tapes"

- AWS tape is a binary blob simply on big long disk file
- Data structures in the file revealed by 3 part headers
  - 3 parts: 2-bytes, 2-bytes, 2-bytes
  - first two bytes are the number of data bytes to follow
  - second two bytes are the number of data bytes in the previous block of data. A “tape” may be read backward
  - the third two bytes, the second byte is reserved and is X'00‘ the first byte is an indicator byte
- HET format unique to Hercules similar to AWS but it allows for compression


```
Particulars of the AWS Format for "Tapes"

Simulating a tape with "one big long disk file" is based on
six-byte "headers" which precede and follow all of the data.

The headers are broken into 3 parts: 2-bytes, 2-bytes, 2-bytes

    1.  The first two bytes are the number of data bytes to
        follow, until the beginning of the next six-byte header.

        Data representation.  The two bytes of the "half-word"
                              are reversed.

        For example, 80-bytes is halfword X'0050'.

        The data representation for the first 2-byte header
        segment for 80-bytes is:  X'5000' with the order of
        the two bytes reversed.

    2.  The second two bytes are the number of data bytes in
        the previous block of data.  The reason for this is so
        you can read the "tape" backward.

    3.  From the third two bytes, the second byte is reserved
        and is X'00'.

        The first byte can be:

        X'80' - the following data chunk is the first chunk
                of the data block

        X'20' - the following data chunk is the last chunk
                of the data block

        X'40' - the previous data chunk represents the last
                block in the file

All of this has implications.  We just have to learn how to look
at it.

HET format, for Hercules, is related to AWS format, but it allows
for compression of the data between the headers, using either the
ZLIB or BZLIB public compression formats.  A Hercules system can
read a HET-format "tape" transparently as a tape.
```

Particulars of the AWS Format for "Tapes" - 2

- For a program to read an AWS tape file it must process the data structures in the file

For example, the headers for a file consisting of the VOL1, HDR1, and HDR2 labels, followed by a tape mark are:

X'5000',X'0000',X'A000'	header before the VOL1 label
X'5000',X'5000',X'A000'	header before the HDR1 label
X'5000',X'5000',X'A000'	header before the HDR2 label
X'0000',X'5000',X'4000'	header after the HDR2 label

- Other examples in the notes

Particulars of the AWS Format for "Tapes" - 2

To read a "tape" in AWS format, you merely need to be able to bounce from one header to the next one, accurately. If you can go through all the headers without an inconsistency, then you can "read the tape".

For example, the headers for a file consisting of the VOL1, HDR1, and HDR2 labels, followed by a tape mark are:

```
    X'5000',X'0000',X'A000'        header before the VOL1 label
    X'5000',X'5000',X'A000'        header before the HDR1 label
    X'5000',X'5000',X'A000'        header before the HDR2 label
    X'0000',X'5000',X'4000'        header after the  HDR2 label,
                                   indicating end-of-file
```

This is immediately followed by another header, which indicates that a data file is to follow. For example:

X'D07F',X'0000',X'A000' which says:

X'7FD0' or 32720 bytes is in the next block of this file.
X'0000' or 0 bytes was in the previous block.
X'A000' means this 32720 bytes is the entire block of data, in one chunk.

The AWS header for a second tape mark would be:

X'0000',X'0000',X'4000'

Particular Tools to Use

- Free Tools
- Tape copying and tape mapping tools. COPYMODS from CBT File 229 does both!
- COPYMODS #229 Multipurpose Copy and Map Tool
- COPYFILE, COPYSLNL, COPYNLNL
- TAPEMAP # 299 Maps tapes and tells a lot
- TCOPY, TAPESCAN, SS0104, TAPECOPY
- PUTXREF, SMPUPD

Particular Tools to Use

My main purpose here is to show you how to use free tools.

There are tape copying tools and tape mapping tools.
Sometimes both functions are in the same program.

I took a tape copying tool, and made a tape mapping tool also
out of it. This is the COPYMODS program, from CBT File 229.

The reason you can do that, is that both tape mapping tools
and tape copying tools read the same tape information. It's
just a question of how much of the tape information you want
to report about.

List of Some Tape Tools:

COPYMODS	-	CBT File 229	Multipurpose Copy and Map Tool
COPYFILE	-	CBT File 229	Copies SL to SL tape files
COPYSLNL	-	CBT File 229	Copies SL to NL tape files
COPYNLNL	-	CBT File 229	Copies NL to NL tape files
TAPEMAP	-	CBT File 299	Maps tapes and tells a lot
TCOPY	-	CBT File 193	Copies or Maps tapes
TAPESCAN	-	CBT File 102	Maps, copies, tells a lot
SS0104	-	CBT File 266	Old program - measures footages
TAPECOPY	-	CBT File 174	Copies tapes - quite versatile
PUTXREF	-	CBT File 118	What's on a PTF tape, by FMID
SMPUPD	-	CBT File 118	Divides a PTF tape into PTFs

Of course, there's IBM's MVS DITTO, and Innovation's FATAR.

I'm going to concentrate on the free stuff, because there aren't
too many forums to tell you how to use those programs.

COPYFILE, COPYSLNL, COPYNLNL

- Original program from which the others were derived was COPYFILE
- COPYFILE supports SL tape copying
- Header and Trailer labels
- COPYSLNL & COPYNLNL


```
COPYFILE, COPYSLNL, COPYNLNL

The original program was COPYFILE.  The others were derived
from COPYFILE.

COPYFILE copies SL files from one tape to another.  SL files
are always in bunches of three.

These are HEADER LABELS, the data file itself, TRAILER LABELS.

Trailer labels have the same stuff as header labels, except
for the block counts in the EOF1 label.

Normal SL tapes come in bunches of 3 files, and COPYFILE counts
on that fact.  (COPYMODS doesn't.)

COPYSLNL strips off the labels when copying tape files.  So the
result of copying from an SL tape is an NL tape.

COPYNLNL doesn't care if the files are labels or not.  Its
original intent was to be able to intelligently divide a long
NL CBT Tape into two.  COPYNLNL always stops after two tape
marks.
```

COPYFILE Control Cards

- COPYFILE, COPYSLNL, and COPYNLNL select files using control cards

For example, if you want to copy files 7, 8, and 22 from

SL tape VOL001 to files 5, 6, and 7 of SL tape VOL002, the

control cards would read:

//SYSIN DD *

7/5 8 22

- More examples in notes show power and flexibility

COPYFILE Control Cards.

The way COPYFILE, COPYSLNL, and COPYNLNL select files, is
through the use of control cards.

For example, if you want to copy files 7, 8, and 22 from
SL tape VOL001 to files 5, 6, and 7 of SL tape VOL002, the
control cards would read:

```
//SYSIN   DD *
   7/5  8   22
```

If you want to copy all of the files after file 22 from
VOL001 to files 8, 9, etc of VOL002, you code:

```
//SYSIN   DD *
   7/5  8   22   ALL
```

If you want to copy all of the files after file 22 from
VOL001 to files 8, 9, etc of VOL002, but you want to stop
after file 50 of VOL001, you code:

```
//SYSIN   DD *
   7/5  8   22   -50
```

The aim of the control cards is to return two numbers to
the calling program. After the two numbers are returned,
the COPYFILE (etc.) program attempts to position both the
output tape and the input tape to the correct place.

Control card operation is now done by an independent routine
called SELCARDS.

Tape positioning has now been adjusted, so you can go backwards
on both the input tape and the output tapes. If you are about
to write over an output file for the second time, you get a
return code 4, and a nasty looking message.

TAPEMAP

- Comes from UCLA
- Several authors including Leonard Woren
- TAPEMAP looks inside files to produce the SYSPRNT2 DD report
- Formats supported: IEBCOPY, FDR, IEBUPDTE, IEBUPDAT, SMPPTFIN, CBT973, embedded PDS, etc.

TAPEMAP

Comes from UCLA. Has several authors. Leonard Woren was one of them. Leonard maintains his own version of TAPEMAP.

The version that was released to the public many years ago by UCLA was played with by many people, including me. Dave Cole put in a bunch of enhancements in 1984. Dave Cole put in 64K block support.

The main "extra" of TAPEMAP is in its SYSPRNT2 DD report, and in the fact that TAPEMAP reports the formats of many kinds of tape files.

```
   Formats supported:  IEBCOPY, FDR, IEBUPDTE, IEBUPDAT,
                       SMPPTFIN, CBT973, embedded pds'es
                       etc.
```

I think Leonard's version supports some different formats, but you have to check with him.

The TAPEMAP program can read a CBT Tape and tell you what member names are in all of the files and imbedded pds'es.

TAPEMAP - 2

- TAPEMAP's two reports are very useful and general
- SYSPRINT DD report shows all the general information
- SYSPRNT2 shows all the specific information


```
  TAPEMAP  -  2

  TAPEMAP's two reports are very useful and general.

  The SYSPRINT DD report shows all the general information
  about the tape, physical results of the scan, label
  information, and if TAPEMAP recognizes the file's format,
  you get to see that too.

  The SYSPRNT2 shows all the specific information on each
  tape file that was written in a "special format".

  EXAMPLE:    SYSPRINT REPORT for SL

VOL=C466MU  OWNER=Sam Golob      CART ANALYSIS PROGRAM   (T A P E M A P)

-------------------------------------------------------------------------
RELOAD   FILE                     PSWD                    INFO
FORMAT   SEQ#  DATASET NAME       REQD  C-DATE  E-DATE  SOURCE  RECFM  L
-------------------------------------------------------------------------
           1  FILE0001                 04.207  98.000  LABELS  FB
CBT DOC    2  VER 466  07-25-04                          SCAN   F

           2  FILE0002                 04.207  98.000  LABELS  FB
           5                                            SCAN   F

           3  FILE0003                 04.207  98.000  LABELS  FB
           8                                            SCAN   F

           4  FILE0004                 04.207  98.000  LABELS  FB
          11                                            SCAN   F

           5  FILE0005                 04.207  98.000  LABELS  FB
          14                                            SCAN   F

           6  FILE0006                 04.207  98.000  LABELS  VB
CBT973    17                                            SCAN   V

  The right side of the report contains number of blocks, lrecl,
  block size and footages, for the particular file, and cumulatively
  for the entire tape up to that point.
```

TAPEMAP - 3

- SYSPRNT2 report from TAPEMAP
- If IEBCOPY is the format, you get DCB info about the original PDS, as well as all the member names
- If SMPPTFIN is the format, you get the PTF numbers and APAR numbers, etc.
- Works for both SL and NL tapes
- PARMs generally subtract information while the defaults mostly “tell all”


```
  TAPEMAP  -  3

  EXAMPLE:  The SYSPRNT2 report from TAPEMAP

FILE0683          (FILE 0683) IS A CBT973 COMPRESSED FILE:   A-ADD   C-C
               FOLLOWING MEMBERS UNLOADED:
               A-$$$#DATE A-$README  A-@FILE683 A-PROC     A-REXX
*********************************************************************
FILE0684          (FILE 0684) IS A CBT973 COMPRESSED FILE:   A-ADD   C-C
               FOLLOWING MEMBERS UNLOADED:
               A-$$$$DOC  A-$$$#DATE A-$$IEFU83 A-$CLINE   A-$CVTIME  A-
               A-#CVIP    A-@FILE684 A-DUMPSMF  A-EDIT     A-IEFACTRT A-
               A-MSGFLUSH A-MSGJOBLG A-MSGNOJLG A-MSGNOLOG A-PDSSCAN  A-
VOL=C466MU  OWNER=Sam Golob      CART ANALYSIS PROGRAM   (T A P E M A P)
---------------------------------------------------------------------
               A-TCPU84   A-UPO      A-UPRINT
*********************************************************************
FILE0685          (FILE 0685) IS A CBT973 COMPRESSED FILE:   A-ADD   C-C
               FOLLOWING MEMBERS UNLOADED:
               A-$$$#DATE A-$README  A-@FILE685 A-CNTL     A-EXEC     A-
*********************************************************************
     *** EOV ***

  If IEBCOPY is the format, you get DCB info about the original pds,
  as well as all the member names.

  If FDR is the format, you get the datasets that were dumped.

  If SMPPTFIN is the format, you get the PTF numbers and APAR
  numbers, etc.

  Works for both SL and NL tapes.

  PARMs generally detract from TAPEMAP operation.  The defaults
  are (mostly) to "tell all".
```

TAPESCAN

- TAPESCAN is a very handy tool
- TAPESCAN probably used to be the best tool for tape analysis till I made the extensive enhancements to COPYMODS
- TAPEMAP gives better overall file information
- TAPESCAN can copy tapes
- TAPESCAN does not dive into files to report on contents

TAPESCAN

TAPESCAN is a very handy tool for tape analysis. Until I made my extensive enhancements to the COPYMODS program, TAPESCAN was my main analysis tool for an unknown tape.

TAPEMAP is better to give you the overall file information on a tape, and to give you an indication about what data is where. TAPESCAN shows more of the "physicality" when you need to know specific information about a tape's file structure and data.

TAPESCAN can copy tapes too. TAPEMAP can't copy tapes.

TAPESCAN formats the labels and gives a hex print of the first 100 bytes of the first nnn records on each tape file. Default is 4 records. If the tape is SL, TAPESCAN will give you a "tape VTOC" at the end of the report, based on the tape label information.

TAPESCAN does not tell you the format of the tape files. That is TAPEMAP's specialty.

TAPESCAN - 2

- TAPESCAN Report
- TAPESCAN 5.2 -------- TAPE ANALYSIS AND COPYING PROGRAM 07/30/04


```
  TAPESCAN  -  2

  Here is the beginning of a sample TAPESCAN report.  Please note
  that it is more "hodgepodgey" looking than the TAPEMAP reports.

TAPESCAN 5.2  --------  TAPE ANALYSIS AND COPYING PROGRAM  07/30/04

OPTIONS IN EFFECT: LIST006,HEX,COUNT,SUMMARY,NOCOPY,NOEOVMOD,VOLSER
OPTIONS IN EFFECT: SKIPEOV000,SKIPTM000,MAXEOV001,MAXTM767,OPT000,NOVTOCONLY
VOL1V466MT                                Sam Golob
EDDFEFFFDE4444444444444444444444444444444E894C999844444444444444444444444444
5631546643000000000000000000000000000000000214076362000000000000000000000000
     VOLUME SERIAL NUMBER=V466MT       OWNER INFORMATION='Sam Golob '
                                        ********************************
                                        ********************************
                                        ** DATASET SEQUENCE NUMBER 0001
                                        ********************************
                                        ********************************
HDR1FILE0001         C466MU00010001      004207 9800000000000IBM OS/VS 370
CCDFCCDCFFFF444444444CFFFDEFFFFFFFF444444FFFFFF4FFFFFFFFFFFFCCD4DE6EE4FFF
849169350001000000000034664400010001000000004207098000000000092406215203 70
     DSNAME=FILE0001           CREATION DATE=07/25/04  EXPIRATION DATE=01
HDR2F327200008000SBGOLOB1/COPY1       B   00000
CCDFCFFFFFFFFFFFECCDDDCF6CDDEF4444444C444FFFFF444444444444444444444444444
849263272000080002276362113678100000002000000000000000000000000000000000
     RECFM=FB        BLKSIZE=32720       LRECL=00080              TRTCH
     CREATED BY JOB SBGOLOB1 IN STEP COPY1
TAPEMARK NO. 0001    BLOCK LENGTHS:  MIN=00080  MAX=00080  AVG=00080
//MVSMODS1 JOB 527TEC000S0003,TEC,CLASS=8,MSGCLASS=5,PRTY=10,
66DEEDDCEF4DDC4FFFECCFFFEFFFF6ECC6CDCEE7F6DECCDCEE7F6DDEE7FF6444444444444
1145246421016205273530002000 3B353B33122E8B42733122E5B7938E10B000000000000
```

TAPESCAN - 3

- TAPESCAN report, showing some of the "tape VTOC" information.

```
TAPEMARK NO. 2099    BLOCK LENGTHS:  MIN=00052  MAX=00052
  AVG=00052

….

VOLUME TABLE OF CONTENTS FOR V466MT    9 TRACK    Sam Golob

SEQ.  DATA SET NAME     RECFM  LRECL  BLKSIZE DEN TRTCH  MAX BLK
  MIN BL

0001  FILE0001           FB    00080   32720   38K  STD   32720
  06400

0002  FILE0002           FB    00080   32720   38K  STD   21040
  21040
```


TAPESCAN - 3

Here is the end of the TAPESCAN report, showing some of the "tape VTOC" information.

```
TAPEMARK NO. 2099    BLOCK LENGTHS:  MIN=00052  MAX=00052  AVG=00052
EOF1FILE0700         V466MT00010700      004207 980000000001IBM OS/VS 370
CDCFCCDCFFFF444444444EFFFDEFFFFFFFF444444FFFFFF4FFFFFFFFFFFFCCD4DE6EE4FFF
5661693507000000000005466430001070000000000420709800000000019240621520370
     DSNAME=FILE0700          CREATION DATE=07/25/04  EXPIRATION DATE=01
EOF2V327160009400SBGOLOB1/NULFILE     B   00000
CDCFEFFFFFFFFFFFECCDDDCF6DEDCCDC44444C444FFFFF444444444444444444444444444
566253271600094002276362115436935000002000000000000000000000000000000000
     RECFM=VB        BLKSIZE=32716       LRECL=00094               TRTCH=
TAPEMARK NO. 2100    BLOCK LENGTHS:  MIN=00080  MAX=00080  AVG=00080
TAPEMARK NO. 2101 -- EOV NO. 001
LENGTH ESTIMATE=1496 FEET 11 INCHES ASSUMING DEN=5 AND TRTCH=STANDARD
(LENGTH ESTIMATE USUALLY ACCURATE WITHIN PLUS OR MINUS TEN PERCENT;  ALMO
SUCCESSFUL PROCESSING OF THIS TAPE COMPLETED:    TOTAL BYTES READ=00324753585
TAPESCAN 5.2  --------  TAPE ANALYSIS AND COPYING PROGRAM  07/30/04   11
VOLUME TABLE OF CONTENTS FOR V466MT    9 TRACK    Sam Golob
SEQ.  DATA SET NAME     RECFM  LRECL  BLKSIZE DEN TRTCH  MAX BLK  MIN BL
0001  FILE0001           FB    00080   32720   38K  STD   32720    06400
0002  FILE0002           FB    00080   32720   38K  STD   21040    21040
0003  FILE0003           FB    00080   32720   38K  STD   32720    22240
0004  FILE0004           FB    00080   32720   38K  STD   32720    02400
0005  FILE0005           FB    00080   32720   38K  STD   32720    22480
```

SMPPTFIN PROCESSING TOOLS

- File 118 of the CBT Tape
- PUTXREF reports on SMP/E elements
  - Many different output forms
  - PDS 8.5 can be used to augment PUTXREF
- SMPUPD can break up a PUT tape
  - PARM=READ can be used to just report

SMPPTFIN PROCESSING TOOLS

These are on File 118 of the CBT Tape.

All of the functionality from all the programs, has been combined into two programs that are written in ASSEMBLER.

These are:

PUTXREF and SMPUPD

PUTXREF tells you which FMIDs your PTFs, APARs, SYSMODs, or USERMODS belong to. Some SYSMODs don't have owning FMIDs, but if they do, PUTXREF will tell you.

SMPUPD will break your PUT tape or other SMPPTFIN-format file into the individual PTFs. It's still kludgey, but it works. SMPUPD can also be run PARM=READ, and it will show you how many card images are in each PTF.

PUTXREF has quite a few different output forms, and it even allows you to use PDS 8.5 to supply ISPF stats for each PTF in the SMPPTS file. It is very useful to be able to look into your SMPPTS ISPF member list screen and sort it by FMID, etc. You need more directory blocks in your SMPPTS dataset. PDS 8.5 will easily enable you to expand this number.

COPYMODS - The new tape "Battlewagon"

- COPYMODS was just a tape copier
- COPYMODS has been expanded to be a “tape do everything” tool!
- Remember tapes are one-dimensional
- Developing COPYMODS needed to know
  - EXCP op codes
  - Tape formats
  - A few more things including assembler


```
COPYMODS  -  The new tape "Battlewagon"

COPYMODS was originally a tape copying program and no more.

But COPYMODS has been expanded almost to be a "tape do
everything" tool.

The reason is because tapes are one-dimensional.  Almost all
tape diagnosing programs have to read the tape.  If you read
the tape, you have access to all the information on it.

Only one COPYMODS function does not read the tape:  LBDQUICK
That is because LBDQUICK was developed to dump the tape labels
from a tape that has "too many" data blocks, like a million,
so it "fast forwards" through the tape's data files.

In developing COPYMODS functionality, we need to know:

    A.  Relevant EXCP op codes that position and
        manipulate a tape.

    B.  What information can be (and is) contained
        in the tape labels.

    C.  What information is obtained by COPYMODS
        when COPYMODS is actually reading the input
        tape.
```

COPYMODS Functionality

- COPYMODS does most everything for tape copying except file selection
- COPYFILE, COPYSLNL, and COPYNLNL support copying with file selection
- COPYMODS has many control parameters
  - Review the source for a general purpose technique you can use to handle adding PARM support to a program cleanly


```
COPYMODS Functionality.

I basically need two programs for tape copying:

COPYMODS does most of it, except for file selection.
I use COPYFILE, COPYSLNL, and COPYNLNL for that.

The reason is that COPYFILE picks selected files
in bunches of three, by counting.  COPYMODS' design
is such that it "feels" when a label is there, and
it doesn't usually consider a tape file to be a
"label" by virtue of its position.

But someday, I might figure out how to incorporate
COPYFILE's selectivity into COPYMODS.

COPYMODS does practically everything else.  And it
can INIT tapes too.  You can copy as many as 16 tapes
at a time, with COPYMODS.

COPYMODS has many controls.  These are incorporated
into the program through the table-driven parm
scanning program called PARMCHEK that is assembled
together with the main COPYMODS code.  If you want
to learn about the many COPYMODS options from the
source code, you should always start with the PARM
table, at label PARMTABL.

Then do something like:  FIND  "PARMFLG6,X'02'"
under ISPF edit or browse, once you've found the parm
bit that's associated with the keyword you're
interested in finding out about.
```

COPYMODS JCL

```
//READ    EXEC  PGM=COPYMODS,REGION=4096K
//STEPLIB  DD  DISP=SHR,DSN=SBGOLOB.W$$.LINKLIB
//SYSPRINT DD  SYSOUT=*
//PARMREPT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//IN     DD VOL=SER=CBT466,DISP=OLD,UNIT=562,
//           LABEL=(,BLP,EXPDT=98000)
//OUT16 DD VOL=SER=CBU466,DISP=OLD,UNIT=563,
//           LABEL=(,BLP,EXPDT=98000)
```



```
  COPYMODS JCL

  Here is default JCL to run COPYMODS without PARM or SYSIN
  modification, and to copy a tape.

//SBGOLOBL  JOB (ACCT#),S-GOLOB,
// NOTIFY=&SYSUID,
// CLASS=B,MSGCLASS=X
//*
//READ   EXEC  PGM=COPYMODS,REGION=4096K
//STEPLIB  DD  DISP=SHR,DSN=SBGOLOB.W$$.LINKLIB
//SYSPRINT DD  SYSOUT=*
//PARMREPT DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//IN     DD VOL=SER=CBT466,DISP=OLD,UNIT=562,LABEL=(,BLP,EXPDT=98000)
//OUT16 DD VOL=SER=CBU466,DISP=OLD,UNIT=563,LABEL=(,BLP,EXPDT=98000)

  You can add PARM input in the EXEC card, or SYSIN input.
  Both are automatically recognized by the COPYMODS program.

  Result of the JCL in SYSPRINT:

COPYMODS PROGRAM - EXTENDED TAPE COPY PROGRAM - LEVEL 080 - OPTIONS DISPLAY
OPTIONS IN EFFECT:

LTMCOPY  BLKCNT    NOHDR1   NOHDR2   NOEOF1   NOEOF2   NOEOV1   NOEOV2
NOLABLDU NOSYSIN   NOOUTVOL NOLABADD NOCHGVOL VOLLBL   NOEOVCHG WRITE
NOPRADDL CORRBLKS  EXNULL   NOLBLFIX NOINITVO NOLIMFIL NOBYTES  NOCUMTOT
NOLIMLAB OPTION    NOSUPPWT NOLBDQUI NOCUMSEP NOSTRIP  NONLLIM  NOSLLIM
NOIDRCOF NOHEXPRT  UNEXWTO  EBCDIC   NOINASC3 NOINASC4 NOENABLT
NOSECOFF NOFOOTAGE NOFOOTDI NOCODEDP NOMINMAX NORECSIZ

 *   THE LIMLABEL OPTION IS ONLY SET BY THE LABELIMIT=   SYSIN CARD
 **  THE LIMFILE OPTION IS ONLY SET BY THE  FILELIMIT=   SYSIN CARD
 *** THE OUTVOL OPTION IS ONLY SET BY THE   TAPEOWNER=   OR OUTVOLAL
COPYMODS has not had to correct any of the options as coded in the JCL.
NOTE - The HEXPRT option displays only original records before they have
       been changed by the COPYMODS program.

BLOCK COUNTS in EOF1 and EOV1 are being corrected for TAPE COPY
                                operation, when outputs are SL.
FILE 00001 CONTAINS 0000142 DATA BLOCKS
FILE 00002 CONTAINS 0000001 DATA BLOCKS
FILE 00003 CONTAINS 0000006 DATA BLOCKS
FILE 00004 CONTAINS 0000002 DATA BLOCKS
FILE 00005 CONTAINS 0000003 DATA BLOCKS
FILE 00006 CONTAINS 0000004 DATA BLOCKS
FILE 00007 CONTAINS 0000041 DATA BLOCKS
    *  *  *  *  *  *  *  *  *  *  *
FILE 00698 CONTAINS 0000001 DATA BLOCKS
FILE 00699 CONTAINS 0000001 DATA BLOCKS
FILE 00700 CONTAINS 0000001 DATA BLOCKS
FILE 00701 CONTAINS 0000000 DATA BLOCKS
COPY CORRECTLY COMPLETED FOR OUT16
```

COPYMODS PARM TABLE

- Flexible parm table managed by the PARMCHEK subprogram
- Keywords associated with bit settings
- Defaults
- If parameters are specified multiple times the last specification is used
- QUANTITY keywords are entered only in SYSIN i.e. OUTVOLALL=volser


```
COPYMODS PARM TABLE

COPYMODS keywords are translated into option bit used by the
program.  The means is the PARMCHEK subprogram, which is
driven by a PARM TABLE.  The parm table is flexible.

A.  Keywords can be up to 8 bytes long.
    You code how many characters are to be searched.

B.  Each keyword is associated with one or more of 80
    possible bits.  These are called bytes PARMFLG1 to
    PARMFLGA, and you can set any bit of each of these
    bytes, on or off.  A flag in the parm table indicates
    if the required bits are to be set ON or OFF.

C.  There is a "default bit" in the parm table.

D.  In the PARMCHEK routine, the PARM TABLE is read
    three times:

        1.  First, to set defaults on or off.

        2.  Second, to read the PARM words from the EXEC
            card PARM field.

        3.  Third, to scan the SYSIN DD name cards, if
            they exist.

Earlier settings are overridden by later settings.

QUANTITY keywords are entered only in SYSIN, by means
of special keywords, such as:

OUTVOLALL=volser      (up to 6 characters)
TAPEOWNER=owner name  (up to 10 characters, 14 for ASCII)
FILELIMIT=nnn         (numeric)
LABELIMIT=nnn         (numeric)
PRINTRCDS=nnn         (numeric)

Sometimes setting a quantity keyword will set off an option
bit that will appear in the OPTIONS report of SYSPRINT.
Some of those cases are enumerated in the OPTIONS report
itself.

Here is part of the PARM TABLE itself:

PARMTABL CSECT
         DC    X'07',X'00',C'LTMSKIP ',X'2000000000000000000'
         DC    X'07',X'00',C'SKIPLTM ',X'2000000000000000000'
         DC    X'07',X'11',C'COPYLTM ',X'2000000000000000000'
         DC    X'07',X'01',C'LTMCOPY ',X'2000000000000000000'
         DC    X'07',X'00',C'LBLINFO ',X'DE000000000000000000'
         DC    X'06',X'00',C'PRINTL  ',X'DE000000000000000000'
         DC    X'06',X'00',C'LABELS  ',X'DE000000000000000000'
```

LABLDUMP option of COPYMODS (and LABADDIN)

- COPYMODS can not just copy tape labels it allows you to manipulate them
- Note: PARM options can be either from EXEC card PARM= field or SYSIN DD statement
- Labels can be peeled off a tape to save, study or for editing
- Labels can easily be added to an NL tape


```
  LABLDUMP option of COPYMODS (and LABADDIN)

  Sample JCL that shows what you can do.  Please notice how
  COPYMODS options can be introduced either in the EXEC card
  PARM field or from the SYSIN DD card.

  For LABADDIN you use the same JCL, except that the LABLDUMP
  DD card becomes a LABADDIN DD card, and the LABADDIN PARM or
  SYSIN keyword has to be coded.  In that case, the input tape
  has to be an NL tape.  It could be the output tape of this
  STRIP step.  LABADDIN is the reverse process of STRIP and
  LABLDUMP.

//SBGOLOB1  JOB ,S-GOLOB,
// NOTIFY=&SYSUID,
// CLASS=B,MSGCLASS=X
//*****************************************************************//
//*   CREATE NL TAPE AND DUMP LABELS                                *//
//*****************************************************************//
//*
//COPY    EXEC PGM=COPYMODS,REGION=4096K,PARM=LABELS
//STEPLIB  DD  DISP=SHR,DSN=SBGOLOB.W$$.LINKLIB
//LABLDUMP DD  DISP=SHR,DSN=SBGOLOB.CBT.CNTL(C466LAB1)
//SYSPRINT DD  SYSOUT=*
//PARMREPT DD  SYSOUT=*
//IN   DD VOL=SER=V466MT,DISP=OLD,UNIT=568,LABEL=(,BLP,EXPDT=98000)
//OUT1 DD VOL=SER=CBT466,DISP=OLD,UNIT=563,LABEL=(,BLP,EXPDT=98000)
//SYSIN  DD *
HEXPRT
STRIP
CUMSEP
LABLDUMP
FOOTAGE
/*

  Here are some of the labels that were dumped.

VOL1V466MT                              Sam Golob
HDR1FILE0001         V466MT00010001     004207 980000000000IBM OS/VS 370
HDR2F327200008000SBGOLOB1/COPY1       B   00000
----ENDOFLABELHEADER
EOF1FILE0001         V466MT00010001     004207 980000000142IBM OS/VS 370
EOF2F327200008000SBGOLOB1/COPY1       B   00000
----ENDOFLABELTRAILER
HDR1FILE0002         V466MT00010002     004207 980000000000IBM OS/VS 370
HDR2F327200008000SBGOLOB1/COPY2       B   00000
----ENDOFLABELHEADER
EOF1FILE0002         V466MT00010002     004207 980000000001IBM OS/VS 370
EOF2F327200008000SBGOLOB1/COPY2       B   00000
----ENDOFLABELTRAILER
  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
HDR1FILE0698         V466MT00010698     004207 980000000000IBM OS/VS 370
HDR2V327160009400SBGOLOB1/NULFILE     B   00000
----ENDOFLABELHEADER
EOF1FILE0698         V466MT00010698     004207 980000000001IBM OS/VS 370
```

INITVOLS, INASC3, INASC4 options of COPYMODS

- COPYMODS can initialize tapes
- INITVOLS
- When is an INPUT tape required to create initialized tapes?


```
  INITVOLS, INASC3, INASC4  options of COPYMODS

  COPYMODS can INIT tapes.  COPYMODS can also copy initted SL
  tapes with nothing on them.

  But to INIT a tape, using the INITVOLS keyword, COPYMODS
  does not need to have an input tape.  In other words, no
  //IN DD  needs to be coded in the JCL.

  This is because an initted SL EBCDIC Tape has only two
  label records, a VOL1 and a special HDR1, and these can
  easily be stored in the program and put out when needed.

  Here is what the labels of an initted SL tape look like.
  They have been dumped with the COPYMODS LABLDUMP option.

VOL1VOL466                                SAM GOLOB
HDR1000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
----ENDOFLABELHEADERINIT

  Here is an initted ASCII LEVEL 4 tape, produced by the
  INASC4 option of COPYMODS.  The character representation
  of the label is in ASCII, and has been translated for
  the purpose of this illustration.

VOL1ASC002                            SAM GOLOB                                  4
HDR1000000000000000000000000010001000100 00000 00000 000000IBMZLA
----ENDOFLABELHEADERINIT

  Note that the difference between an ASCII LEVEL 3 tape
  and a LEVEL 4 tape is the 3 or 4 in column 80 of the
  VOL1 label.  IBM (EBCDIC) VOL1 standard labels don't have
  this distinction.
```

CHGVOL, Block Count correction, SECOFF, IDRCOFF

- COPYMODS can recognize and CHANGE label fields when ordered to do so
- CHGVOL, and OUTVOLALL=outvol
- Block Counts are corrected by default
- SECOFF can turn off security indicators in a copy of a tape
- IDRC indicators can be turned off which is useful if a tape is to be recreated and read on a tape drive which does not support compression


```
CHGVOL, Block Count correction, SECOFF, IDRCOFF

COPYMODS has a very exquisite ability to recognize and change
label fields, when it is ordered to do so, via PARM or SYSIN
keywords.

CHGVOL, and OUTVOLALL=outvol

Normal operation of COPYMODS is to copy all labels as is, from
the input tape to all output tapes.  But sometimes, you want
the output tape to have a different volser.

If you code the CHGVOL keyword, then the output volumes (if they
are SL) will have the volser specified in the JCL.  A RDJFCB is
done on the output DD names, and the volsers are copied into the
program and plugged into the VOL1 labels.

If OUTVOLALL=outvol is coded in a SYSIN card, then all output
volumes will get the same volser, which is the "outvol" volser
that was coded after the OUTVOLALL=  SYSIN keyword.

CORRBLKS and BLKCNT keywords.

These keywords make sure that BLOCK COUNTS in the EOF1 labels
of the output SL tapes are correct.  COPYMODS has read the
input tape, and "knows" what the correct block count for each
tape file should be.  So in order that MVS can read the output
tape and not hiccup, the correct block counts are inserted into
the EOF1 labels before the output tape is written.

This is the default.

This is also done during a LABADDIN operation.  BLKCNT does
the operation during a normal copy, CORRBLKS does it for a
LABADDIN operation.  By default, both keywords are set on.

SECOFF and IDRCOFF.

Tape labels can contain security information, for READ only,
or NO ACCESS.  The SECOFF keyword turns these indicators off
for the copied tapes.

If a tape is produced on a 3490 volume with IDRC compression,
then a character "P" is placed in a few of the label fields
to indicate that.  When MVS tries to read the tape on a drive
that doesn't support compression, it barfs.  Therefore in the
copied tapes, the IDRCOFF option will nullify these "P"
characters wherever they occur, so the output tapes can be
read on a 3480 or 3420 real or virtual tape drive.
```

COPYMODS. File and Label Limiting

- COPYMODS cannot do everything (yet) that COPYFILE, COPYSLNL, and COPYNLNL can
- FILELIMIT=nnn
- LABELIMIT=nnn

COPYMODS. File and Label Limiting

Even though you can't do file selection (yet) with COPYMODS like you can do for COPYFILE, COPYSLNL, and COPYNLNL, you can still cut off the tape copy operation after a certain number of files have been copied. You do that with the SYSIN cards:

FILELIMIT=nnn and
LABELIMIT=nnn

FILELIMIT=nnn will stop copying the input tape after nnn files have been copied. FILELIMIT=nnn will force a tape rewind on the input tape. nnn is multiplied by 3 if a SL tape is the input tape. nnn is used without change, if an NL tape is the input tape. You can force the multiplication by 3, using the SLLIM keyword. You can force NO multiplication by 3, using the NLLIM keyword.

LABELIMIT=nnn is for LABADDIN operations, where you are sandwiching labels around each data file of an NL tape.

If there are more labels than files, then the default is to continue writing null SL files using the extra label sets. The NOEXNULL keyword nullifies this action, and writing of the output tapes will stop after there are no data files left, even if there are more label files left.

LABELIMIT=nnn will always stop writing the output tapes after nnn label sets have been read in.

COPYMODS LBDQUICK label dumping

- Normally COPYMODS will dump all the labels on an SL tape
- IEBTBLK (CBT File 229) was written to produce a test case of a million or more tape blocks
- LBDQUICK reads only the labels not the data
- LBDQUICK implies READ only


```
COPYMODS.  LBDQUICK label dumping.

Normally COPYMODS will dump all the labels on an SL tape
to the LABLDUMP DD name, if the LABLDUMP keyword is coded.

Suppose an SL tape has an enormous number of blocks, and
it takes a long time to read them.  I wrote a program called
IEBTBLK (included in CBT File 229) which can write a million
or more tape blocks on one tape file.  (This was to test the
high order block count field in EBCDIC SL tape labels.)

In order to dump the labels without reading the tape data,
I forward space past the data files, until I hit the label
files, and then I dump the labels.  This is called the LBDQUICK
option of COPYMODS.

LBDQUICK implies READ only.  It's obvious, because if you're
not reading the tape data blocks, how can you copy them?
```

COPYMODS READ only operation

- COPYMODS can modify data but you can use it in READONLY mode and be comfortable you won't
- Setting the write protect on a tape is still not a bad idea
- Keyword parm of READ or READONLY causes COPYMODS to read the tape from the //IN DD name and no //OUT DD names are required


```
  COPYMODS.  READ only operation.

  COPYMODS can be run with a keyword of READ or READONLY.

  This causes COPYMODS to read the tape from the //IN DD name,
  and no //OUTx  DD names are required.

  This allows measuring the input tape, mapping the input tape,
  and HEXPRT of the input tape records, to see what is on them.

  Other keywords helpful in mapping a tape, even though they
  also work during a tape copy operation:

  BYTES, CUMTOT, CUMSEP, LABELS, RECSIZE, MINMAX, HEXPRT.

  Sample output from such a run (without HEXPRT for clarity):

THIS RUN OF THE COPYMODS PROGRAM IS READ-ONLY. JCL VOLSER: V466MT
VOL1 LABEL HAS BEEN DETECTED - VOLSER = V466MT     OWNER = Sam Golob
File 00001 Record number: 0000000001 Size of record: 0000000080
HDR1 LABEL: 1STVOL= C466MU DSN=FILE0001   DS SEQ #: 0001 VOL SEQ #: 0001
            BLK CNT:     000000 CRDATE: 004207 EXDATE:  98000
File 00001 Record number: 0000000002 Size of record: 0000000080
HDR2 LABEL: RECFM=FB   BLOCK LENGTH: 32720  RECORD LENGTH: 00080
            TAPE DEN: 0   JOB/JOB STEP: SBGOLOB1/COPY1      IDRC? NO
File 00001 Record number: 0000000003 Size of record: 0000000080
File 00001 Min Blksize:       80 Max Blksize:       80
FILE 00001 CONTAINS 0000003 DATA BLOCKS CONTAINING 240 BYTES  CUM LABEL:   240
File 00001 Ft @ 6250:        0.1 Cumulative Feet for this Tape:    0.1
File 00002 Record number: 0000000001 Size of record: 0000032720
File 00002 Record number: 0000000002 Size of record: 0000032720
File 00002 Record number: 0000000003 Size of record: 0000032720
File 00002 Record number: 0000000004 Size of record: 0000032720
  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
File 00002 Record number: 0000000141 Size of record: 0000032720
File 00002 Record number: 0000000142 Size of record: 0000006400
File 00002 Min Blksize:     6400 Max Blksize:    32720
FILE 00002 CONTAINS 0000142 DATA BLOCKS CONTAINING 4,619,920 BYTES
                                    CUM DATA : 4,619,920
File 00002 Ft @ 6250:       65.7 Cumulative Feet for this Tape:   65.8
```

COPYMODS HEXPRT option

- Look inside the data for it's true nature
- HEXPRT option can show you the first 132 bytes in hex
- Similar to what TAPESCAN produces


```
  COPYMODS HEXPRT option.

  COPYMODS can show you the first 132 bytes in hex, of the first
  nnn blocks of a tape file.  You use the HEXPRT option to invoke
  the service, and you override the default of nnn=4 by using the

  PRINTRCDS=nnn

  SYSIN card.

  This is similar to what the TAPESCAN program produces
  for hex print.

  You get 4 lines of output by default.

  Example:  (part of a SYSPRINT report where HEXPRT is on)

    COPYMODS has not had to correct any of the options as coded in the JCL.
NOTE - The HEXPRT option displays only original records before they have been
       been changed by the COPYMODS program.

Option READONLY has been forced by not having any OUTxx DDNAMES coded in the JCL
LABELS ARE BEING DUMPED.  OUTPUT DSN=SBGOLOB.CBT.CNTL
                                 MEMBER=VOL466L1 VOLUME=DATA02
THIS RUN OF THE COPYMODS PROGRAM IS READ-ONLY. JCL VOLSER: VOL466
HEXPRT OPTION IS ON IN DEFAULT MODE.      4 RECORDS WILL BE HEX PRINTED.
            TO OVERRIDE, CODE PRINTRCDS=nnn in SYSIN.
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
VOL1VOL466                              SAM GOLOB
EDDFEDDFFF44444444444444444444444444444444ECD4CDDDC44444444444444444444444444444444
5631563466000000000000000000000000000000021407636200000000000000000000000000000000
VOL1 LABEL HAS BEEN DETECTED - VOLSER = VOL466     OWNER = SAM GOLOB
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
HDR1000000000000000000000000000000000000000000000000000000000000000000000000000000
CCDFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
8491000000000000000000000000000000000000000000000000000000000000000000000000000000
FILE 00001 CONTAINS 0000002 DATA BLOCKS
TAPE READ CORRECTLY COMPLETED FOR DDNAME IN, VOLUME SERIAL VOL466


  Interpretation of the characters is EBCDIC by default.  You can
  override this to ASCII.  If an ASCII (AL) tape is being read, then
  the interpretation gets forced to ASCII after the first record.
  That is because HEXPRT reads the raw record before it is
  initially processed by COPYMODS, and its ASCIIness is detected.
```

The COPYMODS MANUAL

- CBT File #229
- Member #MANAUL is an IEBUPDTE PDS with documentation
- If you want to unleash the power of COPYMODS spend the time to read the manual


```
The COPYMODS MANUAL

On File 229, there is a member called #MANUAL.  This is an
IEBUPDTE (really PDSLOAD) formatted pds, which can be
converted to a real pds by the #$MANUAL member, which contains
the requisite JCL.

Much documentation is in this member, to explain the use of
the COPYMODS keywords, and the philosophy behind them.

These are the current members of the COPYMODS.MANUAL pds,
as of LEVEL 080 of COPYMODS:

$INTRO      #CSECTS     #DEFAULT    #GENERAL    #HISTORY    #SYNONYM
#SYSIN      @@PARMS     @PARMTBL    ASCII       BLKCNT      BYTES
CHGVOL      CODEDPRM    EOV2EOF     FOOTAGE     HEXPRT      INASCII
INITVOLS    LABADDIN    LABELS      LABLDUMP    LBDQUICK    LBLFIX
LTM         LTMSKIP     MINMAX      NLLIM       OPTION      PARMCHEK
PRADDLBL    READ        RECSIZE     REWIND      SECOFF      STRIP
SUPPWTO

If you're serious about tape manipulation using COPYMODS, read
the sections of this manual......
```

A Little More about AWS-format TAPES

- AWS format tapes are portable they can be moved to a PC
- AWS format tapes are streamed data not “blocked” the “blocked” structure is only imposed while on an MVS system
- P390 and Hercules emulators can read AWS tapes as though they were real tapes
- VTT2DISK, VTT2TAPE, VTT2CNVT utilities
- AWSUTIL, HETUTL, AWSSL


```
A Little More about AWS-format TAPES

AWS format tapes can be moved to a PC.

AWS format tapes are one continuous stream of data.  It is
only on an MVS system that this data has to be "blocked".

P/390 machines, and Hercules machines can read AWS tapes
as though they were real tapes.

The VTT2DISK and VTT2TAPE  (and VTT2CNVT) programs allow
you to make and handle AWS-format tape files on a "pure MVS"
system, but not to read them as though they were tapes.

Using VTT2TAPE, you can go from a FB-80 folded AWS tape file
and cut a real tape, on a real MVS tape drive.  I think this
is the only way you can cut a real tape from an AWS tape, on
MVS.  That is why I wrote VTT2CNVU.  That will take a VB AWS
tape and convert it into the FB-80 folded format, so you can
use VTT2TAPE to cut a real tape from it.

There are other programs on the CBT Tape which produce
AWS-format tapes on an MVS machine (to read them as tapes,
you have to FTP them to a PC, and use a P/390 or a Hercules
machine.  They are:

   AWSUTIL from Brandon Hill  -  CBT Tape File 477
       This program produces VB-format AWS tapes.

   HETUTL from Leland Lucius  -  CBT Tape File 267
       Leland Lucius is the inventor of the HET format.
       But with no compression, HET equals AWS format.

   AWSSL from Reed Petty  -  CBT Tape File 585
       AWSSL can produce AWS or HET format tapes on
       MVS, directly from MVS datasets.
```

SUMMARY

- Tapes are “one-dimensional”
- Mastery of a few tools makes all the difference
- Go to http://www.cbttape.org and get the Tools!
- Feel free to contact me by email at sbgolob@cbttape.org after SHARE

SUMMARY

As simple as tapes seem to be, being one-dimensional and all,
they have their details.

By spending some time mastering the structure of tape labels
and the op codes which can manipulate a tape drive, you can
appreciate the small intricacy involved in diagnosing a tape.

By mastering a few tape tools, you can become a far better
"data jockey" and you can solve many tape-related problems
for your installation.

Go to www.cbttape.org for more CBT Tape information.

My email : sbgolob@cbttape.org

Good luck!