

# The FATSCOPY How-To-Build-A-Job Guide

V4.9L29

FATSCOPY is a proprietary product of

**INNOVATION DATA PROCESSING**  
**Innovation Plaza**  
**275 Paterson Avenue**  
**Little Falls, NJ 07424**  
**(973)890-7300**



and is available for license exclusively from INNOVATION DATA PROCESSING.

FATSCOPY is a trademark of INNOVATION DATA PROCESSING and is registered with the US Patent Office.

© Copyright 2015 INNOVATION DATA PROCESSING

Revised January 2015

## Introduction

This Guide will step you through the process of constructing a single FATSCOPY batch job to achieve a specific task. **It is not intended to be read start-to-finish.** We will ask you step-by-step questions about the details of your task. As you answer the questions and click on links, you will be guided through the tutorial where we explain which FATSCOPY keywords you will need to add to your batch jobstream.

(If you need to go back to where a link took you from, use **ALT-Left Arrow**.)

We will cover the FATSCOPY parameters needed for most tasks. To create another job for a different task, you should go through the tutorial again from the beginning.

**IMPORTANT: This tutorial assumes that you are running FATSCOPY with its delivered defaults. Your system programmer may have changed the defaults for some keywords, in which case you may need to specify (or not specify) values for those keywords differently than in this tutorial.**

**The symbol ✎ is used to mark keywords which may have had their defaults changed, and require special care.**

For example: there is a parameter called TMSDATA. The delivered default value is TMSDATA=COPY. In this tutorial, we tell you to add a TMSDATA= parameter only if you want to change the value to NOCOPY. But if your site's installation has changed the default to NOCOPY, then you would need to add the TMSDATA= parameter if you wanted the value to be COPY.

This is not a reference document. If you want to look up precisely how a specific keyword works or what the limits are for its parameters, you should refer to the FATSCOPY User Manual.

After you have finished this tutorial, you may wish to refer to some additional FATSCOPY resources which are available.

- the FATSCOPY page on the INNOVATION web site, at <http://www.innovationdp.fdr.com/products/fatscopy/>
- the FATSCOPY User's Manual, which contains documentation on all keywords and describes their parameters
- FATSCOPY ISPF panels, which allow you to create and submit FATSCOPY jobs without directly coding JCL and control statements. These panels are described in section 34.0 of the User's Manual, and further described in the FATSCOPY Quick Start Guide.
- an online demonstration of using FATSCOPY batch jobs and ISPF panels, at <http://www.innovationdp.fdr.com/portfolio/products/demo.cfm> (click on "DEMOS")
- the FATSCOPY Concepts Guide

The User's Manual, Concepts Guide, and Quick Start Guide are available by request by sending an email to [support@fdrinnovation.com](mailto:support@fdrinnovation.com).

## Introduction

You can create a FATSCOPY job to copy one file, one volume, or hundreds of files or volumes. When you are getting ready to set up a FATSCOPY job, you should think about the criteria you will use to specify what input files you want to copy and where you want the output files to go.

- Which do you have: a list of *volumes* you want to copy, or a list of *file names* (or filename patterns)?
- Do you want to *limit* the files selected by certain criteria such as expiration date, creation date, file location, file sequence number, creating job name, last-used job name, data set size, etc.?
- Where do you want the output files to go? Do you want to stack them onto as few tapes as possible? Do you want to add them to a tape which already has files on it?
- Do you want the output files to be cataloged? Should they be renamed?
- Do you want to expire the input files after they have been copied?

To use FATSCOPY, you must have a supported tape management system on your z/OS system. FATSCOPY currently supports IBM's DFSMSrmm, CA Technologies' CA 1 and TLMS, and ASG's Zara. FATSCOPY depends on tape management information for some of the data set selection process and to obtain scratch volumes for the output files. FATSCOPY will update most of the tape management information for the output files, which can be copied from the tape management information for the input files.

FATSCOPY will copy all types of tape files whose locations are not recorded in external databases. Some common types of files are DB2 data files, SMF logs, DFDSS datasets, and ABR backups. In addition, FATSCOPY can copy certain special types of files, such as HSM migration files and ABR migration and application files, and update their databases to reference the copied files. Other special files, which are recorded in external databases, are not good candidates for copying with FATSCOPY at this time. Some of these include CA-SAR, SAS, CA-Dispatch, and Tivoli Storage Manager archive tapes. Some special files which are recorded in external databases *can* be copied with an "image copy", which is an exact bit-for-bit copy of an input tape including the volume serial number; OAM tapes can be copied with an image copy. Contact INNOVATION if you have questions about a specific file type.

You use FATSCOPY control statements to specify what input files you want to copy (or simulate copying), where the output files should go, and what should be done with the input files after a successful copy. A FATSCOPY job:

- always contains either a SIM or COPY control statement (to tell FATSCOPY how to produce the output files, and some miscellaneous details);
- usually contains one or more SELECT control statements (to specify what input files you want to copy);
- may contain EXCLUDE control statements (to further limit the selections);
- may contain a RENAME control statement (to change the names of the output files).

We'll first show you examples of three complete FATSCOPY jobs. You don't need to understand the details in these examples. They are here just to illustrate some of the things you can do with FATSCOPY, and what some typical complete jobs look like.

**You can find more examples in the JCL library provided with FATSCOPY, which you can use as models for building jobs.**

**Preview Example 1: Simple SIM**

A “**Sim**” job performs a simulation, which is where FATSCOPY goes through the data set selection process you specify, so that you can see which data sets FATSCOPY would actually copy in a Copy job with the same selection parameters specified.

```
//JATSAMP1 JOB (J), 'SAMPLE SIM', CLASS=P, MSGCLASS=X
//FATSSIM EXEC PGM=FATSCOPY, REGION=0M
//STEPLIB DD DSN=FATSCOPY.LIBRARY, DISP=SHR
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SELRPT DD SYSOUT=*
//COPYRPT DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//DSNTABLE DD DSN=JAT.FATSCOPY.RESTART, SPACE=(TRK,(15,5),RLSE),
// UNIT=SYSALLDA, DISP=(,CATLG)
//SYSIN DD *
SIM EXPIRED=YES, MULTIFILE=YES, CHECKPT
SELECT ALLDSN, VOL=CCR009
SELECT ALLDSN, VOL=CCR001, CRDATE.GE.2010001
/*
```

In this example,

- EXPIRED=YES means both expired and unexpired files are eligible for selection. MULTIFILE=YES means that tapes containing multiple files will be eligible for selection.
- All data sets from volume CCR009 will be selected.
- Any data sets on volume CCR001 which were created on or after January 1, 2010 will be selected (CRDATE.GE.2010001).
- A report about the data sets selected or bypassed will be written to the SELRPT data set.
- A report showing what the copy results using these SELECTs would look like will be written to the COPYRPT data set.
- Any error messages will be written to the ERRORRPT data set.
- The optional CHECKPT keyword and DSNTABLE DD statement causes a disk data set named JAT.FATSCOPY.RESTART to be allocated and cataloged. The selection results of this simulation are saved in this data set; it can be used in a “Restart” job to perform a copy of these data sets without re-specifying all of the FATSCOPY control statements.

No tapes are actually mounted, and no files are copied. The SELRPT data set shows the selection results, and the COPYRPT data set contains a report on the simulated copy results. [Click here to see what some sample Select and Simulation Reports look like.](#)

## Introduction

### Preview Example 2: Simple COPY

A “Copy” job actually copies data sets. You might want to “stack” data sets from several underutilized tapes onto a single tape or migrate data sets from one type of tape device to another.

```
//JATSAMP2 JOB (J), 'SAMPLE COPY', CLASS=P, MSGCLASS=X
//FATSCOPY EXEC PGM=FATSCOPY, REGION=0M
//STEPLIB DD DSN=FATSCOPY.LIBRARY, DISP=SHR
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SELRPT DD SYSOUT=*
//COPYRPT DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//TAPEOUT DD DSN=DUMMY, UNIT=VT90, DISP=(NEW, KEEP)
//SYSIN DD *
          COPY EXPIRED=NO
          SELECT CATDSN=JAT.TEST. **
/*
```

In this example,

- EXPIRED=NO means that files that are marked as “expired” in the tape management system will not be copied.
- All cataloged tape data sets with the first two index levels “JAT.TEST” in their data set names will be selected.
- A report about the data sets selected or bypassed will be written to the SELRPT data set.
- A report on the copy results will be written to the COPYRPT data set.
- Any error messages will be written to the ERRORRPT data set.

[Click here to see what a sample Copy Report looks like.](#)

**Preview Example 3: Another COPY**

This shows an example of a COPY with more complex selection criteria, and more parameters specifying where you want the output files to go. This shows an example of how you can “stack” data sets onto as few output tapes as possible.

```
//JATSAMP3 JOB (J), 'SAMPLE COPY',CLASS=P,MSGCLASS=X
//FATSCOPY EXEC PGM=FATSCOPY,REGION=0M
//STEPLIB DD DSN=FATSCOPY.LIBRARY,DISP=SHR
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SELRPT DD SYSOUT=*
//COPYRPT DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//AUDIT DD DSN=JAT.FATSCOPY.AUDITDSN,UNIT=SYSALLDA,
// DISP=(NEW,CATLG),SPACE=(TRK,(5,5))
//SYSABEND DD SYSOUT=*
//TAPEOUT DD DSN=DUMMY,UNIT=VT90,DISP=(NEW,KEEP)
//SYSIN DD *
COPY CAT=NO,EXPDTGROUP=999,MULTIFILE=NO,
EXPIRED=NO
EXCLUDE CATDSN=JAT.TEST.DEPT2.**
EXCLUDE CATDSN=JAT.TEST.*.SALES.**
SELECT CATDSN=JAT.TEST.**
/*
```

In this example,

- The output files will not be cataloged (CAT=NO). You might use this when you are making copies for a backup site, and want the original files to remain cataloged on your system.
- All the output data sets will be “stacked” onto as few tapes as possible (EXPDTGROUP=999).
- Only data sets on single-file tapes will be eligible for copying (MULTIFILE=NO); if they’re already “stacked”, they won’t be copied.
- Files that are marked as “expired” in the tape management system will not be copied (EXPIRED=NO).
- To be selected, the first 2 index levels of a data set name must be “JAT.TEST”. Furthermore, the EXCLUDE statements (which must precede the SELECT statement) specify that any data set with a 3<sup>rd</sup> index level “DEPT2” will not be selected, nor will any with a 4<sup>th</sup> index level “SALES”. So,
  - JAT.TEST.DEPT3.PAYROLL will be selected
  - ABC.TEST.DEPT3.PAYROLL will not be selected (1<sup>st</sup> index level doesn’t match)
  - JAT.TEST.DEPT2.PAYROLL will not be selected (due to 1<sup>st</sup> EXCLUDE)
  - JAT.TEST.DEPT3.SALES will not be selected (due to 2<sup>nd</sup> EXCLUDE)
- A report about the data sets selected or bypassed will be written to the SELRPT data set.
- A report on the copy results will be written to the COPYRPT data set.
- Any error messages will be written to the ERRORRPT data set.
- The optional AUDIT DD statement causes a disk data set named JAT.FATSCOPY.AUDITDSN to be allocated and cataloged. A record of the files copied is saved in this data set. The FATAUDIT program, which is provided with FATSCOPY, can be used to display a readable Audit Report.

We will now go through a series of **Questions** to help guide you through the process of constructing your own FATSCOPY batch job. Text in lowercase shows where you have to substitute your own value.

## DD Statements, and COPY or SIM

If you've already set up DD statements in a previous job and want to reuse them, and you know whether you are going to use SIM or COPY, you can [click here](#) to start setting up your FATSCOPY control statements. (If you need to go back to where a link took you from, use **ALT-Left Arrow**.)

After your JOB statement, start with the following DD statements:

```
//jobname EXEC PGM=FATSCOPY,REGION=0M
//SYSPRINT DD SYSOUT=*
//TAPESUMM DD SYSOUT=*
//SELRPT DD SYSOUT=*
//COPYRPT DD SYSOUT=*
//ERRORRPT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSIN DD *
/*
```

### Q: Is FATSCOPY in the linklist on your z/OS system?

*No* – **Add** the following STEPLIB statement (or change it to JOBLIB if appropriate) after the EXEC statement:

```
//STEPLIB DD DSN=your.fatscopy.library,DISP=SHR
```

You will need a SIM or COPY control statement immediately after the SYSIN statement. For now, it is just the word **SIM** or **COPY**. You will later add parameters to this statement. The first parameter is separated from **SIM** or **COPY** by a **blank**; additional parameters are **separated by commas**. If you have more parameters than you can fit on a single statement, end the statement with a comma and continue adding parameters on the next statement (as shown in the Copy preview example above).

### Q: Do you want to *simulate* copying data sets, or perform an actual *copy*?

*Simulate* – **Add the SIM control statement.** So, now the last 3 lines of your jobstream are

```
//SYSIN DD *
SIM
/*
```

Then [click here to skip ahead](#).

*Copy* – **Add the COPY control statement.** So, now the last 3 lines of your jobstream are

```
//SYSIN DD *
COPY
/*
```

Also **add** the following TAPEOUT DD statement before the SYSIN DD:

```
//TAPEOUT DD DSN=DUMMY,UNIT=devtype,DISP=(NEW,KEEP)
```

where `devtype` is the type of device you want the output files to be copied to. Then [click here to skip ahead](#).

When you are doing a Sim to select data sets but not actually copy them at this time, you can *save* the selection results in a “DSNTABLE” data set; this disk data set can be used by a subsequent Copy job to perform the actual copy without having to re-specify the selection control statements. This also saves some processing time during the Copy job.

**Q: Do you want to create a data set which saves the results of this simulation for a later Copy job?**

*No* – [click here to skip ahead](#).

*Yes* – **Add CHECKPT** to your SIM statement.

Also, **add** a DSNTABLE DD statement before the SYSIN DD:

```
//DSNTABLE DD DSN=your.dsntable.filename,SPACE=(TRK,(p,s),RLSE),  
//          UNIT=SYSALLDA,DISP=(,CATLG)
```

Use any valid name for the DSNTABLE data set. Typical values for the disk data set primary and secondary allocation sizes: p=15, s=5. Then [click here to skip ahead](#).



**Q: Do you want to do a “Restart” to perform a copy of the data sets which were selected by a previous Sim job which used a DSNTABLE DD statement?**

Yes – Add **RESTART** to your COPY statement.

Also, add a DSNTABLE DD statement:

```
//DSNTABLE DD DSN=your.dsntable.filename,DISP=SHR
```

where the DSNTABLE data set name is the same as that used in the SIM job you want to restart.

You can tell FATSCOPY to maintain an audit record of all the copies it does during this job. You can either create a new audit file for this job or append it to an existing audit file. The audit records can be saved in a sequential disk file, or in a system logger file. Only one copy job can access a sequential disk file at a time. If you want multiple jobs to be able to write audit records to the same audit file at the same time, you must use the system logger. Only the sequential disk file method is described in this How-To Guide; for information on using the system logger for saving audit records, see “Writing Audit Records” in section 32.6 of the FATSCOPY User’s Manual.

If all your FATSCOPY jobs append to the same audit file, you can have a record of all files which have been copied on your system. Or, you might want to have a different audit file for each week, or each month, or each department. You can later use the FATAUDIT program, which is provided with FATSCOPY, to create a human-readable Audit Report. [Click here to see an example of a FATAUDIT Audit Report](#). (See the FATSCOPY User Manual for directions on how to use FATAUDIT.)

**Q: Do you want to create (or add to) a sequential Audit data set to keep a record of what files have been copied?**

No – [click here to continue](#)

Yes – To create a *new* Audit data set, **add** this DD statement before the SYSIN DD:

```
//AUDIT DD DSN=your.audit.dataset,UNIT=SYSALLDA,
// DISP=(NEW,CATLG),SPACE=(TRK,(p,s))
```

Use any valid name for the AUDIT data set which is not already cataloged. Typical values for the disk data set primary and secondary allocation sizes “p” and “s”: 5 for each.

Or, to add to an *existing* Audit data set, **add** this DD statement:

```
//AUDIT DD DSN=your.audit.dataset,UNIT=SYSALLDA,DISP=MOD
```

where the AUDIT data set name is the name of the existing cataloged audit data set you want to append to. Then [click here to continue](#).

## Input Data Set Selection

We will now proceed to some questions to guide you through adding parameters based on which input data sets you want to copy (or simulate copying).

**Q: Are you doing a COPY for a set of data sets previously selected by a SIM where you saved the selections with the CHECKPT parameter (using the DSNTABLE DD statement)?**

Yes – Since you’ve already done all of the input data set selection, [click here to skip ahead](#).

### Choose your *selection method*: CATDSN vs. ALLDSN

There are two basic ways you select the data sets to be copied. You must choose one or the other, and cannot use both in the same FATSCOPY job. You can select either *by data set name or mask* or *by volume serial*. Pick a method based on how you want to specify which data sets should be selected.

Selecting by data set name or mask is a “**CATDSN**” selection. To use this method, all data sets that you want to select *must* be *cataloged* in the z/OS catalog. Choose this one if you know the **names** or pattern of data set names (such as “all datasets whose names start with JAT.DAILY.”) of the cataloged data sets you want to select.

Selecting by volume serial is an “**ALLDSN**” selection. In this method, FATSCOPY queries your tape management system to find all of the files on the volumes you specify. These data sets do not need to be cataloged in the z/OS catalog. Choose this one if you know which specific **volumes** contain the data sets you want to select.

An “image copy” is a special type of copy where an exact bit-for-bit copy, including the volume serial, is made on the output tape. A common use for an image copy is when you are copying files that are recorded in an external data base, and the data base records the volser and file sequence number for each file it controls. OAM datasets can be copied with an image copy without requiring updates to its data base. **If you want to make an image copy, you must use an ALLDSN selection.**

After you start the selection process, you will be able to further modify the selections and specify exclusion criteria to limit what gets selected. For example,

- You can specify that you want to copy all data sets whose names start with JAT.DAILY, but only those which were created more than 7 days ago. Or,
- You can specify that you want to copy volume PRD009, except for those files on that volume which expire before March 1, 2014.

**Q: Do you want to select by *data set name or mask*, or by *volume serial*?**

Select by *data set name or mask* – [click here for CATDSN selections](#).

Select by *volume serial* – [click here for ALLDSN selections](#).

## CATDSN Selections

First, we will go through some limits which will apply to *all* of your selections.

With CATDSN, files which are marked as expired in your tape management system are normally *not* selected.

### **Q: Do you want to allow data sets which are marked as Expired to be selected?**

Yes – Add **EXPIRED=YES** to your COPY or SIM statement.

### ***Files on multifile volumes***

With CATDSN, by default FATSCOPY will *only* select files from single-file volumes. Files on volumes with more than one file will *not* be selected, to avoid copying tapes which have already been stacked.

### **Q: Do you want to allow files from volumes with more than one file on a volume to be selected?** (You will probably want to use this if you are migrating volumes which might be already stacked to a new tape device.)

Yes<sup>x</sup> – Add **MULTIFILE=YES** to your COPY or SIM statement.

### ***Multivolume files***

A multivolume file is one which spans from one volume to another. By default FATSCOPY will copy a file whether or not it spans volumes.

### **Q: Do you want to limit FATSCOPY so that it will *not* select multivolume files?**

Yes<sup>x</sup> – Add **MULTIVOL=NO** to your COPY or SIM statement, then [click here to continue](#).

### **Q: Do you want to limit FATSCOPY so that it will select *only* multivolume files?**

Yes<sup>x</sup> – Add **MULTIVOL=ONLY** to your COPY or SIM statement.

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

**Data set names and masks**

You can specify, one by one, the data sets you want to copy. Or you can specify a “mask” which is a pattern to match. When you specify a mask, FATSCOPY searches your system catalog for all of the data sets which match that mask. **You will build one statement for each data set name or mask.**

A full description of data set masks is in Section 32.5 of the FATSCOPY manual. **You should refer to that section for the full set of rules for constructing masks**, and many examples of varying complexity. Here are a few simple examples of data set name masks:

<b>JAT.MONTHLY.**</b>	selects all cataloged data sets where the first two high-level qualifiers are “JAT.MONTHLY”, and have any number of additional index levels.
<b>JAT.**</b>	selects all cataloged data sets where the first high-level qualifiers is “JAT”; the second level must exist but can have any valid characters; and have any number of additional index levels.
<b>JAT.MONTHLY.*</b>	selects all cataloged data sets where the first two high-level qualifiers are “JAT.MONTHLY”, and have exactly 1 more high-level qualifier.
<b>JAT.MONTHLY.    </b>	selects all cataloged data sets where the first two high-level qualifiers are “JAT.MONTHLY”; the third index level consists of exactly 3 alphabetic characters; and there are no more levels.
<b>JAT.MONTHLY.    *</b>	selects all cataloged data sets where the first two high-level qualifiers are “JAT.MONTHLY”; the third index level consists of at least 3 alphabetic characters; and there are no more levels.
<b>JAT.MONTHLY.JULY**</b>	selects all cataloged data sets where the first two high-level qualifiers are “JAT.MONTHLY”; the third index level starts with “JULY”; and has any number of levels after that.

Some of the key “wildcard” characters are

- “/” or “%” for a single alphanumeric character
- “|” for a single alphabetic character
- “+” for a single numeric character
- “?” for a single national character
- “\*” with other characters to specify any number of characters in that position in that index level
- “\*” by itself for an index level with any characters
- “\*\*\*” for zero or more characters or index levels

**Build your selections**

Now we'll begin to form your individual selections, or continue with the next one.

**Q: Do you want to specify an *individual* data set name, or specify a *mask* to match to select multiple data sets?**

*Individual data set name* – **Add** a new control statement, **SELECT CATDSN=datasetname** immediately after your COPY or SIM statement or after any SELECT statements you've already added, using a fully-qualified data set name.

*Specify a mask* – **Add** a new control statement, **SELECT CATDSN=datasetmask** immediately after your COPY or SIM statement or after any SELECT statements you've already added.

**Limiting selections for this name or mask**

This data set, or all of the cataloged data sets matching the mask you specified, will be selected for copying, unless you *limit* the selection by specifying additional criteria.

**Q: Do you want to select this data set, or these data sets, *without* any further limits?**

Yes – [click here to skip ahead](#).

You will add parameters to your SELECT statement to limit the selection. **These limits apply only to this data set name or mask.** Each SELECT statement can have its own list of limits. Separate each parameter/value set with a comma. If necessary, you may continue the parameter list onto the next line by ending a line with a comma. A data set must satisfy ALL of the limits defined by the parameters on a SELECT statement in order to be selected.

For some parameters, you can specify a *list* of values for the parameter, enclosed in parentheses. For example, for VOL, you could specify VOL=(PRD001,DEV123,ABC999). When you specify a list of parameter values, the data set must match *any one* of the values in order to satisfy the parameter. If you specify only one value, you can omit the parentheses.

For many parameters, you can specify a *single* instance of the parameter with a single operator and value, or a *pair* of the same parameters to specify the beginning and ending of a range. The 5 valid **operators** are = , > , >= , < , <= ; equivalent operators are **.EQ.** , **.GT.** , **.GE.** , **.LT.** , **.LE.** . For example, with FILESEQ, you could specify values such as

```
FILESEQ=6
FILESEQ.GT.5
FILESEQ<3
FILESEQ>7,FILESEQ<=20
FILESEQ.GE.3,FILESEQ.LT.9
```

When you specify a pair of parameters, the data set must satisfy *both* of the conditions in order to be eligible to be selected. (In the last example, files 3 and 8 would be selected, but files 2 and 9 would not.)

**Q: Do you want to want to limit the selection to data sets on specific *volume(s) (or range of volumes)*?**

Yes – Add **VOL=(volser1,volser2,...)** to this SELECT statement, up to 20 volumes; or

Add **VOLoperatorn** to this SELECT statement; or

Add **VOLoperatorn1,VOLoperatorn2** to this SELECT statement for both the beginning and end of a range.

**Q: Do you want to limit the selection to data sets located on specific *device type(s)*?**

Yes – Add **DEVTYPE=(devtype1,devtype2,...)** to this SELECT statement, up to 5 device types.

**Q: Do you want to limit the files selected to specific *file sequence number(s) (or range of numbers)* on the input volumes (or multivolume sets)?**

Yes – Add **FILESEQoperatorn** to this SELECT statement; or

Add **FILESEQoperatorn1,FILESEQoperatorn2** to this SELECT statement for both the beginning and end of a range.

**Q: Do you want to limit the files selected to files with specific *expiration dates (or range of dates)*, as recorded in tape management?**

Yes – Add **EXPDATEoperatordate** to this SELECT statement; or

Add **EXPDATEoperatordate1,EXPDATEoperatordate2** to this SELECT statement for both the beginning and end of a range,

where you specify **dates** in Julian format yyyy.ddd or yyyyddd .

**Q: Do you want to limit the files selected to files with specific *creation dates (or range of dates)*, as recorded in tape management?**

No – [click here to skip ahead](#).

You can specify the creation date(s) as absolute dates (year, and day of year), or a number of days ago relative to the current date.

**Q: Do you want to specify the creation dates as *absolute dates* or as *relative days ago*?**

*Absolute date* – Add **CRDATEoperatordate** to this SELECT statement; or

Add **CRDATEoperatordate1,CRDATEoperatordate2** to this SELECT statement for both the beginning and end of a range,  
where you specify **dates** in Julian format yyyy.ddd or yyyyddd .

*Relative number of days* – Add **CRDAYSoperatorn** to this SELECT statement; or

Add **CRDAYSoperatorn1,CRDATEoperatorn2** to this SELECT statement for both the beginning and end of a range,  
where you specify **n** as the number of days before today. (Example: CRDAYS>4 selects data sets created more than 4 days ago. CRDAYS=0 selects data sets created today.)

**Q: Do you want to limit the files selected to files *created by specific job names (or job name prefixes)*, as recorded in tape management?**

Yes – Add **CRJOB=(jobname1,jobname2,...)** to this SELECT statement

where you specify up to 20 fully-qualified job names or job name prefixes (1-7 alphanumeric characters followed by an asterisk, such as “JAT\*”).

**Q: Do you want to limit the files selected to files with specific *data set sizes (or range of sizes)*, as recorded in tape management?**

Yes – Add **DSSIZEoperatorsize** to this SELECT statement; or

Add **DSSIZEoperatorsize1,DSSIZEoperatorsize2** to this SELECT statement for both the beginning and end of a range,  
where you specify **sizes** in megabytes.

**Q: If you have *RMM, TLMS, or Zara tape management*, do you want to limit the files selected to files with specific *Current Location value(s)*, as recorded in tape management?**

Yes – Add **LOCATION=(location1,location2,...)** to this SELECT statement

where you specify up to 5 fully-qualified Location values.

**Q: If you have *CA 1 tape management*, do you want to limit the files selected to files with specific *Outcode value(s)*, as recorded in CA 1?**

Yes – Add **OUTCODE=(outcode1,outcode2,...)** to this SELECT statement

where you specify up to 5 fully-qualified Outcode values. Each value can be up to 4 characters in length; if a value contains an embedded blank, enclose that Outcode value within single quotation marks.

**Q: If you are selecting GDG datasets, do you want to limit the number of generations which will be selected from a single GDG?**

Yes – Add **CATLIMIDGDG=nnn** or **CATLIMITGDG=-nnn** to this SELECT statement  
where the maximum value of nnn is 255.



This completes this SELECT statement.

**Q: Do you want to specify *another* data set name or data set mask to select?**

No – [Click here to skip ahead.](#)

**Q: Do you want to apply the *same limits* you used on a previous SELECT statement to another data set name or mask?**

Yes – **Add** another SELECT statement by *duplicating* the previous statement, adding it after the previous SELECT statement and changing only the data set name or mask on it. Repeat with more statements if you have a several data set names or masks to select with the same limits.

**Q: Do you want to specify apply *different limits* from those you used on a previous SELECT statement to another data set name or mask?**

Yes – You will build another SELECT statement to specify another set of limits applying only to the new statement. [Click here to go back up to start the new SELECT statement.](#)

After CATDSN is used to select data sets, FATSCOPY normally sorts the data sets by expiration date group, volser, and file sequence number. When this is done, the number of input tape mounts is minimized. It also minimizes the number of output tapes needed. However, if you need the data sets copied in the same sequence as you have specified them in the SELECT control statements, you can turn off this sorting.

**Q: Do you want to turn off the sorting which minimizes the number of tape mounts and output tape volumes needed?**

Yes<sup>x</sup> – **Add SORT=NO** to your COPY or SIM statement (*not recommended*).

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

## CATDSN Exclusions

If you have selected data sets with masks, you can exclude some data sets which would otherwise be selected by specifying a fully-qualified data set name, or another mask, for the datasets to be excluded. For example, you may have specified `SELECT CATDSN=JAT.MONTHLY.**`, but want to copy all the data sets matching that mask *except* for those which *also* match `JAT.MONTHLY.DEPT2*.**` .

To do this, you can now build one or more `EXCLUDE` statements. An `EXCLUDE` statement looks almost like a `SELECT` statement, except that it specifies criteria to exclude data sets which would otherwise be selected by the `SELECT` statements which follow it. For this tutorial, we will put one or more `EXCLUDE` statements before the first (or only) `SELECT` statement, so that they will apply to all of the `SELECT` statements.

**Q: Do you want to specify exclusion criteria that will apply to *all* of the `SELECT` statements?**

No – [click here to skip ahead](#).

**Q: Do you want to specify *an individual data set name* or *a mask* to match to exclude multiple data sets that would otherwise be selected by the `SELECT` statements you build?**

*Individual data set name* – Add a new control statement, **`EXCLUDE CATDSN=datasetname`** immediately after your `COPY` or `SIM` statement, and before your first `SELECT` statement, using a fully-qualified data set name.

*Specify a mask* – Add a new control statement, **`EXCLUDE CATDSN=datasetmask`** immediately after your `COPY` or `SIM` statement, and before your first `SELECT` statement. The rules for forming an exclusion mask are the same as for forming a selection mask.

**Q: Do you want *this* exclusion to apply only to data sets on specific *volume(s)* (or range of volumes)?**

Yes – Add **`VOL=(volser1,volser2,...)`** to this `EXCLUDE` statement, up to 20 volumes; or

Add **`VOLoperatorn`** to this `SELECT` statement; or

Add **`VOLoperatorn1,VOLoperatorn2`** to this `SELECT` statement for both the beginning and end of a range.

**Q: Do you want *this* exclusion to apply only to data sets located on specific *device type(s)*?**

Yes – Add **`DEVTYPE=(devtype1,devtype2,...)`** to this `EXCLUDE` statement, up to 5 device types.

This completes this `EXCLUDE` statement.

**Q: Do you want to specify an exclusion for *another* data set name or mask?**

*Yes* – You will build another EXCLUDE statement, and specify another set of limits applying only to the new statement. (If you want the same limits as the EXCLUDE you just built, you will need to also specify them on the new statement.) [Click here to go back up to start the new EXCLUDE statement.](#)

*No* – [click here to continue.](#)

## ALLDSN Selections

First, we will go through some limits which will apply to *all* of your selections.

With ALLDSN, by default FATSCOPY will select data sets whether or not they are marked as expired in your tape management system.

### **Q: Do you want to *prevent* data sets which are marked as Expired from being selected?**

*Prevent* – Add **EXPIRED=NO** to your COPY or SIM statement.

### ***Multivolume sets***

When a file spans from one volume to another volume, the 2 volumes are a “multivolume set”. If the file at the end of the second volume spans to a third volume, that third volume is also part of the multivolume set. And so on, until a volume ends with a non-spanning file.

When you specify *any* volume which is part of a multivolume set, ***all volumes of the multivolume set are selected***. For example, if you have a multivolume set consisting of volumes PRD001, PRD003, and PRD156, and you specify that you want to select volume PRD003, all three volumes of that set will be selected.

By default, FATSCOPY will select files whether or not they are part of multivolume sets.

### **Q: Do you want to limit FATSCOPY to *not* selecting files that are part of multivolume sets?**

Yes <sup>Ⓜ</sup> – Add **MULTIVOL=NO** to your COPY or SIM statement, then [click here to continue](#).

### **Q: Do you want to limit FATSCOPY to select *only* files that are part of multivolume sets?**

Yes <sup>Ⓜ</sup> – Add **MULTIVOL=ONLY** to your COPY or SIM statement.

### ***How you specify volumes***

You will specify, one by one, the volumes you want to copy. Each tape volume you specify is queried against your tape management database. All files on that input volume will be selected for copying (unless limited by additional operands, which you can add later). If the volume is part of a multivolume tape set, all data sets on all volumes in that set will be selected.

In addition, FATSCOPY supports copying entire *physical* tape volumes from supported ***tape-based virtual tape servers*** (VTS). Currently supported systems are IBM’s TS7740 and 3494, and the Oracle StorageTek VSM virtual tape system.

### **Q: Are you copying from one of these VTS systems?**

No – [click here to skip ahead](#).

---

<sup>Ⓜ</sup> The default value for this keyword may have been changed on your system.

### ***Virtual Tape Server considerations***

In addition to specifying *logical* tape volumes as described above, you can specify individual *physical* volume serials. When you specify a physical volsr, FATSCOPY queries the VTS's internal database to get a list of all the virtual tape volumes on that physical tape. It then works as if you had specified *each* of those logical volumes.

With virtual tape systems, many logical volumes may be located on a single high-capacity physical volume. If you want to copy volumes from a VTS (such as when you are migrating to another system), time savings can be achieved by grouping together all of the logical volumes from a single physical tape to avoid tape mount thrashing.

You can specify a mixture of physical and logical volsers to select.

#### **Q: Will you want to specify *any* physical volsers to copy?**

No – [click here to skip ahead](#).

#### **Q: Which type of VTS system do you have?**

*IBM TS7700 or 3494* – Add **VIRTTYPE=IBM** to your COPY or SIM statement.

Also, **add** the following DD statement to your JCL:

```
//MAPTAPE DD UNIT=vtsunit,DISP=(,CATLG),DSN=any.maptape.filename,  
//          LABEL=(,SL),RETPD=1,  
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,TRTCH=NOCOMP)
```

where `vtsunit` is the virtual tape device *containing the input volumes to be copied*. The MAPTAPE data set is a temporary tape file which will be used by FATSCOPY to retrieve information from that VTS.

*Oracle StorageTek VSM* – Add **VIRTTYPE=STK** to your COPY or SIM statement. No DD statement changes are required.

***Build your selections***

Now we'll begin to form your individual selections, or continue with the next one. **You will build one statement for each physical or logical volser.**

**Q: Did you specify VIRTTYPE=STK or VIRTTYPE=IBM to copy from a supported VTS system?**

*No* – **Add** new control statement, **SELECT ALLDSN,VOL=volser**, immediately after your COPY or SIM statement or after any SELECT statements you've already added, where **volser** is a fully-qualified volume serial. This volume, or all volumes in the multivolume set, will be selected. [Then click here to skip ahead.](#)

You can copy an individual *logical* volume (or range of volumes), or an individual *physical* volume. If you want to copy a physical volume, you can specify which one by either (a) its physical volser, or (b) by the logical volser of one of the logical volumes on that physical tape.

**Q: Do you want to copy an individual *logical volume* (or range of logical volumes), individual *physical volume by physical volser*, or individual *physical volume by logical volser*?**

*Logical volume* – **Add** a new control statement, **SELECT ALLDSN,VOL=volser**, immediately after your COPY or SIM statement, or after any SELECT statements you’ve already added, where **volser** is a fully-qualified *logical* volume serial. This volume, or all volumes in the multivolume set, will be selected.

If you want to copy a *range* of *nnnn* volumes starting with **volser**, add **NUMVOLS=nnnn** to your SELECT statement, where **nnnn** is a value from 1 to 32767.

For example,

```
SELECT ALLDSN,VOL=V00111,NUMVOLS=3
```

gives exactly the same results as specifying these 3 separate SELECT statements:

```
SELECT ALLDSN,VOL=V00111
SELECT ALLDSN,VOL=V00112
SELECT ALLDSN,VOL=V00113
```

If you are copying a large range of volumes, you may want to break this range up into several separate jobs so that you can run smaller jobs in parallel or in a limited processing time window. The FATSCOPY PUNCH keyword provides a way to generate multiple copy jobs starting from a large NUMVOLS= range, saving those jobs in a partitioned data set. An explanation of how to use PUNCH is beyond the scope of this tutorial. Details on how to use PUNCH can be found in section 32.7 of the FATSCOPY User’s Manual, “Writing FATSCOPY Jobs to a Partitioned Data Set”.

*Physical volume by physical volser* – **Add** a new control statement, **SELECT ALLDSN,PHYSVOL=phyvol**, immediately after your COPY or SIM statement or after any SELECT statements you’ve already added, where **phyvol** is a fully-qualified *physical* volume serial. All logical volumes on this physical tape will be selected. If any of them are part of multivolume sets, all volumes in the set(s) will also be selected.

*Physical volume by logical volser* – **Add** a new control statement, **SELECT ALLDSN,ALLPHYS=logvol**, immediately after your COPY or SIM statement or after any SELECT statements you’ve already added, where **logvol** is a fully-qualified *logical* volume serial. FATSCOPY will determine which physical tape contains that logical volume. All logical volumes on that physical tape will be selected. If any of them are part of multivolume sets, all volumes in the set(s) will also be selected.

### **Image copy**

An image copy is an exact bit-for-bit copy of the input tape. The output volume has the same volser as the input volume, with the same files in the same sequence as the input volume. A common use for an image copy is when you are copying a volume, such as an OAM tape, which contains files whose locations are recorded in an external data base, and so the volser for each file must be preserved. The output device for an image copy can be a real cartridge tape or an EMC DLM or MDL virtual tape system. Do not attempt an image copy with other types of virtual tape output devices.

### **Q: Do you want to do an image copy?**

Yes – Add **IMAGE** to your COPY statement, and [click here to skip ahead](#).

### **Limiting selections for this volume (or multivolume set)**

All of the data sets on the selected volume (or group of volumes) will be selected for copying, unless you *limit* the selection by specifying additional criteria.

### **Q: Do you want to select all the data sets on this volume (or group of volumes), without any further volume-specific limits?**

Yes – [click here to skip ahead](#).

You will add parameters to your SELECT statement to limit the selection. *These limits apply only to the data sets on this volume (or group of volumes)*. Each SELECT statement can have its own list of limits. Separate each parameter/value set with a comma. If necessary, you may continue the parameter list onto the next line by ending a line with a comma. A data set must satisfy ALL of the parameters on a SELECT statement in order to be selected.

For some parameters, you can specify a *list* of values for the parameter, enclosed in parentheses. For example, for CRJOB, you could specify CRJOB=(MAINT01,MAINT02,PROD099). When you specify a list of parameter values, the data set must match *any one* of the values in order to satisfy the parameter. If you specify only one value, you can omit the parentheses.

For many parameters, you can specify a *single* instance of the parameter with a single operator and value or a *pair* of the same parameters to specify the beginning and ending of a range. The 5 valid **operators** are = , > , >= , < , <= ; equivalent operators are **.EQ.** , **.GT.** , **.GE.** , **.LT.** , **.LE.** . For example, with FILESEQ, you could specify values such as

```
FILESEQ=6
FILESEQ.GT.5
FILESEQ<3
FILESEQ>7,FILESEQ<20
FILESEQ.GE.3,FILESEQ.LT.9
```

When you specify a pair of parameters, the data set must satisfy *both* of the conditions in order to be eligible to be selected. (In the last example, files 3 and 8 would be selected, but files 2 and 9 would not.)



**Q: Do you want to limit the files selected to specific *file sequence number(s)* (or range of numbers) on the input volumes (or multivolume sets)?**

Yes – Add **FILESEQoperatorn** to this SELECT statement; or

Add **FILESEQoperatorn1,FILESEQoperatorn2** to this SELECT statement for both the beginning and end of a range.

**Q: Do you want to limit the files selected to files with specific *expiration dates* (or range of dates), as recorded in tape management?**

Yes – Add **EXPDATEoperatordate** to this SELECT statement; or

Add **EXPDATEoperatordate1,EXPDATEoperatordate2** to this SELECT statement for both the beginning and end of a range,

where you specify **dates** in Julian format yyyy.ddd or yyyyddd .

**Q: Do you want to limit the files selected to files with specific *creation dates* (or range of dates), as recorded in tape management?**

No – [click here to skip ahead](#).

You can specify the creation date(s) as absolute dates (year, and day of year) or as a number of days ago relative to the current date.

**Q: Do you want to specify the creation dates as *absolute dates* or as *relative days ago*?**

*Absolute date* – Add **CRDATEoperatordate** to this SELECT statement; or

Add **CRDATEoperatordate1,CRDATEoperatordate2** to this SELECT statement for both the beginning and end of a range,

where you specify **dates** in Julian format yyyy.ddd or yyyyddd .

*Relative number of days* – Add **CRDAYSoperatorn** to this SELECT statement; or

Add **CRDAYSoperatorn1,CRDATEoperatorn2** to this SELECT statement for both the beginning and end of a range,

where you specify **n** as the number of days before today. (Example: CRDAYS>4 selects data sets created more than 4 days ago. CRDAYS=0 selects data sets created today.)

**Q: Do you want to limit the files selected to files *created by specific job names* (or job name prefixes), as recorded in tape management?**

Yes – Add **CRJOB=(jobname1,jobname2,...)** to this SELECT statement

where you specify up to 20 fully-qualified job names or job name prefixes (1-7 alphanumeric characters followed by an asterisk, such as "JAT\*").

**Q: Do you want to limit the files selected to files with specific *data set sizes (or range of sizes)*, as recorded in tape management?**

Yes – Add **DSSIZEoperatorsize** to this SELECT statement for a single value; or  
Add **DSSIZEoperatorsize1,DSSIZEoperatorsize2** to this SELECT statement for a range,  
where you specify **sizes** in megabytes.

**Q: If you have RMM, TLMS, or Zara tape management, do you want to limit the files selected to files with specific *Current Location value(s)*, as recorded in tape management?**

Yes – Add **LOCATION=(location1,location2,...)** to this SELECT statement  
where you specify up to 5 fully-qualified Location values.

**Q: If you have CA 1 tape management, do you want to limit the files selected to files with specific *Outcode value(s)*, as recorded in CA 1?**

Yes – Add **OUTCODE=(outcode1,outcode2,...)** to this SELECT statement  
where you specify up to 5 fully-qualified Outcode values. Each value can be up to 4  
characters in length; if a value contains an embedded blank, enclose the Outcode value  
within single quotation marks.

This completes this SELECT statement.

**Q: Do you want to specify *another* volume?**

No – [Click here to skip ahead.](#)

**Q: Do you want to apply the *same limits* you used on a previous SELECT statement to another volume?**

Yes – Add another SELECT statement by duplicating the previous statement, adding it after the  
previous SELECT statement and changing only the volser on it. Repeat with more  
statements if you have several volumes to select with the same limits.

**Q: Do you want to specify apply *different limits* from those you used on a previous SELECT statement to another volume?**

Yes – You will build another SELECT statement, and specify another set of limits applying only to  
the new statement. [Click here to go back up to start a new SELECT statement.](#)

## ALLDSN Exclusions

You can exclude some data sets which would otherwise be selected by specifying a fully-qualified data set name or a data set mask for the datasets to be excluded. For example, you may have specified

```
SELECT ALLDSN, VOL=PRD001
SELECT ALLDSN, VOL=DEV005
```

but want to copy all the data sets on those volumes *except* for those whose names match the mask JAT.MONTHLY.DEPT2\*.\* .

To do this, you can now build one or more EXCLUDE statements. An EXCLUDE statement looks almost like a SELECT statement, except that it specifies criteria to exclude data sets which would otherwise be selected by the SELECT statements which follow it. For this tutorial, we will put one or more EXCLUDE statements before the first (or only) SELECT statement, so that they will apply to all of the SELECT statements.

**Q: Do you want to specify exclusion criteria which will apply to *all* of the SELECT statements?**

*No* – [click here to skip ahead](#).

**Q: Do you want to specify an *individual data set name*, or specify a *mask* to match to exclude multiple data sets which would otherwise be selected by the SELECT statements you have built?**

*Individual data set name* – **Add** a new control statement, **EXCLUDE DSN=datasetname** immediately after your COPY or SIM statement, and before your first SELECT statement, using a fully-qualified data set name.

*Specify a mask* – **Add** a new control statement, **EXCLUDE DSN=datasetmask** immediately after your COPY or SIM statement, and before your first SELECT statements.

**This completes this EXCLUDE statement.**

**Q: Do you want to specify an exclusion for *another* data set name or mask?**

*Yes* – You will build another EXCLUDE statement, and to specify another set of limits applying only to the new statement. (If you want the same limits as the EXCLUDE you just built, you will need to also specify them on the new statement.) [Click here to go back up to start the new EXCLUDE statement](#).

*No* – [click here to continue](#) to the next page.

## More questions about processing your inputs

### Q: Do you want to limit the *total number* of input files that FATSCOPY will select?

Yes<sup>Ⓜ</sup> – Add **MAXTOTFILE=n** to your COPY or SIM statement, where **n** is the maximum number of files to select. (When this number is exceeded, FATSCOPY will stop selecting files. The default value is 1000. The maximum value allowed is 65535.)

You can specify some more exclusion criteria which will globally apply to *all* of your SELECT statements, instead of putting them on each individual SELECT statement or using an EXCLUDE statement.

### Q: Do you want to EXCLUDE *all* datasets *created* by one or more specified *programs* from being selected?

Yes – Add **EXCRPGM=(pgmname1,pgmname2,...)** to your COPY or SIM statement. Up to 10 values are allowed. Each value must be fully qualified.

### Q: Do you want to EXCLUDE *all* datasets that were *last used* by one or more specified *job names* from being selected? (*valid only if your tape management system is RMM, TLMS, or Zara*)

Yes – Add **EXLASTJOB=(jobname1,jobname2,...)** to your COPY or SIM statement. Up to 10 values are allowed. Each value can be fully qualified, or 1-7 alphanumeric characters followed by an asterisk. (For example, FATS\* means exclude all data sets last used by jobs whose names start with “FATS”.)

### Q: Do you want to EXCLUDE *all* datasets that were *last used* by one or more specified *programs* from being selected? (*valid only if your tape management system is RMM, CA 1, or Zara*)

Yes – Add **EXLASTPGM=(pgmname1,pgmname2,...)** to your COPY or SIM statement. Up to 10 values are allowed. Each value must be fully qualified.

### Q: Do you want to allow data sets on tapes that are marked in your tape management system as “*Offsite*” to be selected? (Choose this only if you know the tapes will actually be available for mounting when you do the copy.)

Yes<sup>Ⓜ</sup> – Add **OFFSITE=YES** to your COPY or SIM statement.

---

<sup>Ⓜ</sup> The default value for this keyword may have been changed on your system.

### *Old device types*

When copying older tapes, sometimes they were last read on a device which is no longer available, but they can still be read on a newer device which you have. When this is the case, you need to tell FATSCOPY which type of device it should use for the input tape mounts.

**Q: Are you copying tapes which may have been created or read on a device you don't have any more?**

Yes – Add **UNITIN=devtype** to your COPY statement, where **devtype** is an esoteric (such as “TAPE”), generic (such as “3490”) or specific (such as “3A0”) unit name to be used to mount the input tapes.

### *Special input file types*

**Q: Are you an ABR customer (level 5.4/70 or higher)?**

No – [click here to skip ahead](#).

**Q: Do you want to allow FATSCOPY to select ABR Archive and Application backup data sets?**

No<sup>‡</sup> – [click here to skip ahead](#).

Yes<sup>‡</sup> – Add **ABRARC=YES** to your COPY or SIM statement.

**Q: Are the ABR modules in the linklist?**

No – Add a **STEPLIB** or **JOBLIB DD** statement for the ABR library.

The first index level of ABR Archive and Application backup data sets is saved in the FDR Global Options Table. The index level value is usually “FDRABR”.

**Q: Do you want FATSCOPY to use a different value to identify ABR datasets?**

Yes<sup>‡</sup> – Add **ABRINDEX=yourvalue** to your COPY or SIM statement.

FATSCOPY usually does NOT select HSM backup and migration data sets.

**Q: Do you want to allow FATSCOPY to select HSM Migration Level 2 data sets?**

Yes<sup>‡</sup> – Add **HSMML2=YES** to your COPY or SIM statement. FATSCOPY will be allowed to select and copy HSM ML2 data sets; an HSM RECYCLE command will be used to copy the data sets and update the HSM MCDS.

---

<sup>‡</sup> The default value for this keyword may have been changed on your system.

FATSCOPY identifies data sets as HSM backup and migration data sets by comparing them with the following masks:

```
*.COPY.HMIGTAPE.DATASET
*.HMIGTAPE.DATASET
*.COPY.BACKTAPE.DATASET
*.DMP.*.V*.D++++.T+++++
*.BACKTAPE.DATASET
*.DMP.T+++++.*.D++++.V*
```

**Q: Does your installation use a different mask for HSM backup data sets?**

Yes<sup>x</sup> – Add **HSMBAKMASK=yourbackupmask** to your COPY or SIM statement.

**Q: Does your installation use a *different* mask for HSM migration data sets?**

Yes<sup>x</sup> – Add **HSMMIGMASK=yourmigrationmask** to your COPY or SIM statement.

**Q: Are you doing a SIM, rather than a COPY?**

Yes – You are finished with the input parameters. [Click here to skip ahead.](#)

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

## Scratching Input Files After They Are Copied

After each input file has been copied, FATSCOPY can tell your tape management system to reset the expiration date of the input file to a specified number of days after the current date. If FATSCOPY expires all of the files on a volume in a single job, it can reset the volume expiration date so that the tape can be returned to scratch status after the files expire, making the tape available for reuse.

If you are doing a migration from one set of tapes to another, or from one system to another, you probably want to allow this expiration. If you will still need the input data sets later, you should not use this option.

By default, FATSCOPY leaves the expiration dates intact, unless you specify that the input files be scratched.

**Important note for Zara users: If your installation has the Zara “Scratch by 1st” option set to “Y”, Innovation strongly advises against allowing FATSCOPY to scratch input files. Using this option may result in unintended loss of data.** If FATSCOPY expires the first data set on a tape, the volume will be scratched when “Scratch by 1st” is set to “Y”, even if other non-expired data sets remain on the volume.

### **Q: Do you want to *expire the input files* after they have been copied?**

*No*<sup>x</sup> – click [here to skip ahead](#).

*Yes*<sup>x</sup> – Add **TMSINPUT=SCRATCH** to your COPY statement, unless you are using Zara tape management, in which case you should use **TMSINPUT=ZARASCR** after considering the “Important note” above.

You can choose how many days will be added to the current date to get the new expiration date of the input files. By default, 4 days are added to the current date, but you can pick another value.

### **Q: Do you want to use a *different value* to be added to today’s date to calculate the input expiration date?**

*Yes*<sup>x</sup> – Add **ADDXDAYS=n** to your COPY statement, where “n” is the number of days to add to the current date. **Innovation strongly recommends a value NO LESS THAN 4.**

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

## Output Data Set Processing

Most functions relating to the output files are needed only for Copy jobs, not for Sim jobs. The exception is the set of Grouping functions, where the output groups are calculated during the selection process and saved if necessary for a later Restart.

### **Q: Are you doing a SIM, rather than a COPY?**

Yes – [click here to skip ahead](#) to the Grouping functions.

## Cataloging

The most important consideration for your output data sets is deciding on cataloging. A specific data set name can be cataloged only once in the *OS catalog* (distinct from *tape management*, which can record *multiple* locations for a single data set name).

### **Special note**

If you want both the input files *and* output files to be cataloged, you can tell FATSCOPY to create the output file with a *different data set name*. This “rename” function results in an output file with the same data as the input file, except that the file labels for the output use the new data set name. With two different names, the input file can remain cataloged to the input volume, and the output file can be cataloged to the output volume.

We will discuss *how* to rename your output files later. For now, you need to know that if you specify the RENAME keyword, then *all* of your output files for this job will be renamed with the new name or mask which you specify.

### **Q: Do you want FATSCOPY to *never* catalog an output data set?**

*Never catalog* – Add **CAT=NO** to your COPY statement. (This option *always* leaves the catalog status of the input file *unchanged*.) Then [click here to continue](#).

Remember that the name that FATSCOPY uses to catalog the output data set is the same as the input file name, unless you use RENAME. In that case, the new name you’ve specified may or may not match the name of an existing file name already in the catalog.

### **Q: Do you want FATSCOPY to *always* catalog an output data set, whether or not the name you are trying to catalog is already in the catalog?** (If the input file, or any other file, was already cataloged with this name, that file will no longer be cataloged after the copy.)

*Always catalog* – Add **CAT=RECAT** to your COPY statement, then [click here to continue](#).



## Output Data Set Processing

Otherwise, there are two other options which you can use:

1. Catalog the output *only if* the name you are trying to catalog *is already in the catalog*, AND is cataloged to the input file *which was actually copied* (matching volser and file sequence number).
2. Catalog the output *if* the name you are trying to catalog is *not already in the catalog*. (If the name is already in the catalog, the catalog will not be updated.)

Choice #1 can be used to be sure that the catalog is not inadvertently updated when the “wrong” file is copied. For example, you might have different files named JAT.AUGUST.SALES.REPORT on volumes RP2009, RP2010, and RP2011, with the one on volume RP2011 being the one which is cataloged. If you do an ALLDSN copy of volume RP2009, one of the files which gets copied is JAT.AUGUST.SALES.REPORT, but not the version which is cataloged. With this choice the output file will not be cataloged, so the catalog will still point to the version on volume RP2011.

Choice #2 can be used when you are using RENAME to be sure you haven’t inadvertently picked a name which matches a name of an already-cataloged file, and still want that file to stay cataloged.

### **Q: In which case do you want to allow FATSCOPY to catalog the output file name?**

*Choice 1: Only if it exactly matches existing catalog entry – Add **CAT=ONLY** to your COPY statement. (After the copy, the input file will *never* be in the catalog.)*

*Choice 2: If not already in catalog – Add **CAT=YES** to your COPY statement. (This option *always* leaves the catalog status of the input file *unchanged*.)*

## Tape management data

Tape management data for the output data sets will normally be copied from the tape management information for the corresponding input data sets. This includes fields such as the creation date & time, creating job name, creating step name, and expiration date.

### Q: Do you want to *suppress copying tape management data*?

*Yes*<sup>x</sup> – Add **TMSDATA=NOCOPY** to your COPY statement. The current date will be recorded as the creation date of the output file; the FATSCOPY job/step will be recorded as the creating job/step.

### Q: Do you want to specify an *explicit* data set expiration or retention, instead of using the value from tape management for the input files?

*Yes* – Add the appropriate **EXPDT=** or **RETPD=** parameter to your TAPEOUT DD statement. This value will apply to all of the output files copied.

### Q: Are you using RMM?

*No*—[click here to skip ahead](#).

### Q: Did you use SELECT CATDSN?

*No* – [click here to skip ahead](#).

### Q: Do you want to copy the RMM Owner information from the input volume to the output volume?

*Yes* – Add **COPYOWNER** to your COPY statement. (If data sets from multiple volumes are stacked on a single volume, the Owner field of the volume of the last data set copied is used.)

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

## Renaming your output files

When FATSCOPY copies a file, the output file usually has the same data set name as the input file. The locations (volser and file sequence number) of *both* files will be recorded in your *tape management system*. The location of *only one (or neither)* of the files will be recorded in your *system catalog*.

You can tell FATSCOPY to give the output files *different file names* from the input files. You might do this if you want both the input and output files to be cataloged or if you need a different file name for any other purpose. If you specify the RENAME keyword, then all of the output files for this job will be renamed. You cannot apply the RENAME to just some of the files copied. An output file will contain the same data as the corresponding input file, but will have a different data set name in its file labels. You cannot rename files if you are doing an image copy.

When you do a RENAME, then the catalog status of the input file is always unchanged. The output file may or may not be cataloged. Both the input and output files will be recorded in your tape management system.

### **Q: Do you want the output files to have *different names* than the input files?**

*No* – [click here to continue](#) (the usual case).

When you rename an output file, you change the data set name of the input data set index-level-by-index-level by specifying a mask. You can

- *replace* an index level with a specific value;
- *insert* an index level with a specific value;
- *delete* an index level.

You can do some very complex renaming with the RENAME function. For complete details for complex RENAME masks, **you should refer to section 32.4 of the FATSCOPY User Manual to determine what mask you need.**

To add the Rename function, **add the RENAME control statement** after your last SELECT or EXCLUDE statement. The RENAME statement looks like this:

**RENAME LF=ALL,NEWI=newindexmask**

where **newindexmask** is the RENAME mask which will be applied to all output files.

In the simplest case, FATSCOPY will use each index level specified in the mask you specify in place of the corresponding index in the original name. Any remaining index levels at the end of the name are copied unchanged. This can be used to easily change the first indexes of the name.

Example: if you use NEWI=ABC.XX, the first 2 index levels of all filenames will be replaced to form the output file names.

- The copy of JAT.MONTHLY.DATA will be named ABC.XX.DATA
- The copy of JAT.MONTHLY.DATA.BACKUP would be named ABC.XX.DATA.BACKUP
- The copy of JAT.WEEKLY.DATA would also be named ABC.XX.DATA (probably not what you really want to do!)

**Q: Do you want the Copy Report to display the output file name after the renaming has been done, instead of the name of the input file, for each file copied?**

Yes – Add **SHOWNEWN** to your COPY statement.

## Grouping

You can specify how the data sets will be *grouped* on the output tapes. Each “group” of data sets is used to create a *separate* output volume, or multivolume set.

**Q: Are you doing a RESTART of data sets selected by an earlier SIM?**

Yes – Grouping functions were saved by the SIM, so [click here to skip ahead](#).

**Q: Do you want *all* of the output data sets to be grouped together (“stacked”) in a *single group*? This minimizes the number of output tapes used.**

Yes<sup>x</sup> – Add **EXPDTGROUP=999** to your COPY or SIM statement, then [click here to skip ahead](#).

**Q: Did you select data sets by CATDSN or by ALLDSN?**

*CATDSN* – [click here to continue](#).

*ALLDSN* – [click here to continue](#).

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

### ***Grouping with CATDSN***

When data sets have been selected using CATDSN, the output data sets are normally grouped by the *expiration dates* of the input files. This allows files that will expire in the same range of days to be on a single tape (or multivolume set), so that the tape can be scratched and reused as soon as possible.

You can change this so that the output data sets are instead grouped together the same way they are grouped on the input volumes. All the files from one volume (or multivolume set) will go together to an output volume or multivolume set, regardless of their expiration dates. Use this option if you need to retain the input data set grouping or want to minimize the number of mounts of each input tape.

#### **Q: Do you want to group by *input volume set* instead of the usual Expiration Date grouping?**

Yes – Add **GROUPBY=VOLSET** to your COPY or SIM statement, then [click here to continue](#).

You can specify what the allowed spread of expiration dates will be for a single group. The normal value is 7; data sets that expire within 7 days of each other will form a single group. (So if you have 14 daily data sets, the first 7 will be in the first group, and the next 7 will be in the next group.)

#### **Q: Do you want a group spread of a number of days *different* from 7?**

Yes<sup>x</sup> – Add **EXPDTGROUP=n** to your COPY or SIM statement, where the number of days **n** may be from 0 to 998.

[Then click here to skip ahead](#).

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

### **Grouping with ALLDSN**

When data sets have been selected using ALLDSN, the output data sets are normally grouped together the same way they are grouped on the input volumes. All the files from one volume (or multivolume set) will go together to an output volume or multivolume set, regardless of their expiration dates. This minimizes the number of mounts for each input tape.

You can change this so that the output data sets are grouped by the *expiration dates* of the input files. This allows files which will expire in the same range of days to be on a single tape (or multivolume set). Use this option if you want to scratch and reuse your output tapes as soon as possible.

#### **Q: Do you want to group by *expiration date* instead of the usual grouping by input volume?**

No – [click here to skip ahead](#).

Yes – Add **GROUPBY=EXPDATE** to your COPY or SIM statement.

You can specify what the allowed spread of expiration dates will be for a single group. The normal value is 7; data sets which expire within 7 days of each other will form a single group. (So if you have 14 daily data sets, the first 7 will be in the first group, and the next 7 will be in the next group.)

#### **Q: Do you want a group spread of a number of days *different* from 7?**

Yes<sup>x</sup> – Add **EXPDTGROUP=n** to your COPY or SIM statement, where the number of days **n** may be from 0 to 998.

#### **Q: Is this a SIM job?**

Yes – Other output functions are not needed by a SIM, so [click here to skip ahead](#).

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

## Stacking

### **Q: Do you want to limit the number of datasets that can be copied (“stacked”) on a *single* tape?**

Yes<sup>x</sup> – Add **MAXFILE=n** to your COPY statement, where the number of files **n** may be from 1 to 65535. When this number is exceeded, FATSCOPY will call for a scratch tape and continue copying onto the new tape. If you specify “1”, no stacking is done – you can “unstack” input tapes which had stacked data sets.

When FATSCOPY is stacking files on a tape and reaches the end of the tape, how it proceeds depends on the size of the file fragment at the end of the tape.

- If the partial file is *small*, FATSCOPY will delete the partial file from the end of the tape and re-copy that file starting on a new tape to try to avoid creating a spanned file. (However, it can’t do this if the partial file is already the first file on the tape!)
- If the partial file is *large*, FATSCOPY will continue copying the file onto a subsequent tape, creating a spanned file and a multivolume set.

By default, any partial file smaller than 1000Mb will be deleted and recopied, while any partial file larger than 1000Mb will be continued as a spanned file. You can change the number which divides “small” and “large”.

### **Q: Do you want to *change* the value which determines what FATSCOPY will do when the output end-of-volume is reached?**

Yes<sup>x</sup> – Add **LASTFILE=n** to your COPY statement, where **n** is a number of megabytes. Any time a partial file at the end of a tape is *smaller* than this size, it will be deleted and restarted on a new tape. Otherwise, the file will span to another volume.

If you want to almost always allow a spanned file to be created, use LASTFILE=1. If you want to avoid creating a spanned file whenever possible, use LASTFILE=9999.

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

### ***Appending files to existing tapes***

In the usual, simplest case, a FATSCOPY copy job requests the mount of a fresh scratch tape and starts copying files to that new volume. FATSCOPY jobs are normally independent of each other.

However, you can specify that FATSCOPY should *append* files to a volume that already has data sets on it. There are two ways you can append to an existing volume:

- By explicitly specifying a starting file sequence number FATSCOPY should start with, and exactly which volume you want to write to. This requires you to know how many files are already on that volume.
- By using a special “LASTAPE” data set which is created and cataloged by FATSCOPY to serve as a “last file on tape” marker. When you use this method, you do not need to know which volume should be written to, or how many files are already on it. FATSCOPY will add data sets to the tape written by previous FATSCOPY job.

### **Q: What do you want to do about appending files?**

- To start with a fresh tape, and *not write* a LASTAPE file, [click here](#). (*this is the usual case*)
- To start with a fresh tape, and *write* a LASTAPE file so a future job can add to it, [click here](#).
- To append to an existing tape *without using* a LASTAPE file, [click here](#).
- To append to an existing tape *that contains* a LASTAPE file created by a previous FATSCOPY job, [click here](#).

After all the files have been copied by this job, FATSCOPY will catalog a **LASTAPE** data set after the last file copied. It is an empty data set which is used as a “last file on tape” marker by FATSCOPY. You can specify any valid data set name (which is not already cataloged) for this file. In a future job, when you want to append to this tape, you will specify the same data set name so that FATSCOPY can append to the tape at that point. Different jobs may define different LASTAPE data sets for different purposes (such as JAT.LASTAPE.PAYROLL for your payroll copy jobs, JAT.LASTAPE.SALES for your sales report copy jobs, and JAT.LASTAPE.INVENTORY for your inventory report copy jobs).

**Add LASTAPE=new.lastape.dataset.name** to your COPY statement, using a *new* LASTAPE data set name (any valid data set name which is not already cataloged) for the volume you want to add to. Then [click here to continue](#).



## Output Data Set Processing

To append to the output tape for a previous job which used a particular LASTAPE file, use the same cataloged data set name specified in the LASTAPE= parameter for that job.

**Add LASTAPE=old.lastape.dataset.name** to your COPY statement, using the cataloged LASTAPE data set name for the volume you want to add to.

At the end of this job, *another* empty dataset with the same LASTAPE data set name will be created and the name re-cataloged as a new “last file on tape” marker, which can be used by yet another FATSCOPY job to append more files to the tape.

When LASTAPE has been used, FATSCOPY will normally not allow a multivolume set to be created. If adding a data set to a tape would cause the file to span volumes, it will instead be written to a new scratch tape.

**Q: Do you want to allow FATSCOPY to create spanned output files while LASTAPE is being used?**

*No* – [click here to skip ahead](#).

There are two different ways which FATSCOPY can decide while using LASTAPE whether or not a particular file should be added onto a tape, or should instead start on a new scratch tape:

- Specify a percentage of the tape used. If more than the specified percentage of a tape is used *before* writing a new file, no matter what the size of the new file is, a new scratch tape will be used. (Note that the size of certain types of tape may not be accurately recorded by some tape management systems; also, this option is not allowed with TLMS.)
- Specify the number of bytes for a partial output file. If end-of-tape is reached when writing an output file, and *fewer* than the specified number of bytes have already been written, then the partial file is deleted and the copy of that file is re-started on a fresh scratch tape. If *more* bytes have been written, the output is allowed to span to another volume.

**Q: Do you want to specify a *percentage of tape or bytes of a partial file*?**

*Percentage* – Add **THRESHOLD=n** to your COPY statement, where **n** is a percentage (1 to 99). If more than this percentage has already been used on the output tape, then a new scratch tape is used for the next output file.

*Number of bytes* – Add **LASTMULT** to your COPY statement. Also add **LASTFILE=n** to your COPY statement, where **n** is a number of megabytes. Any time the number of bytes written when the the end of a tape is reached *smaller* than this size, the partial file will be deleted and the copy of that file will be restarted on a new tape. Otherwise, the file will span to another volume.

Then [click here to continue](#).

To append data sets to a tape which already has files on it without using a LASTAPE file, you need to know the volser and the file sequence number of the last file currently on the tape.

- Add **VOL=SER=volser** to your TAPEOUT DD statement; and
- Add **OUTFSEQ=n** to your COPY statement, where **n** is a number one larger than the file sequence number of the last file already on the tape.

## Miscellaneous output parameters

### Q: Do you want the output files to be written to tapes which are *SMS-managed*?

Yes – Do one of the following:

- specify **STORCLAS=storageclass** on the TAPEOUT DD statement, or
- specify a data set name **DSN=SMSmanagedname** on the TAPEOUT DD statement that is assigned to a SMS-managed tape by your SMS ACS routines.

The SMS classes specified or assigned will be used for all data sets copied.

### Q: Do you want the output files to be written with *larger blocks* than were read on the input files?

No – [click here to skip ahead](#).

Yes – Add **REBLOCK=n** to your *COPY statement*. Input files with block sizes less than **n** will be copied to output files with block sizes as close to **n** as possible. Input files with block sizes larger than **n** will not be reblocked.

### Q: Do you want to *prevent* files which block sizes which are already “large enough” from being reblocked?

Yes – Add **MAXRBLK=n** to your *COPY statement*. Input files with block sizes larger than **n** will not be reblocked.

### Q: Do your input files have *user labels*? (uncommon)

No – [click here to skip ahead](#).

Normally, when FATSCOPY copies files, the labels are also copied. However, with certain types of user labels FATSCOPY should *not* be used to copy the labels. For example, some encryption products (such as CA Tape Encryption) write their own user labels that contain the file’s encryption keys.

### Q: Do you want to *suppress (prevent)* FATSCOPY from copying the user labels?

Yes<sup>‡</sup> – Add **SUPULAB** to your *COPY statement*. User labels will NOT be copied.

## Archive Control File backups

### Q: Are you an ABR customer?

No – [click here to skip ahead](#).

---

<sup>‡</sup> The default value for this keyword may have been changed on your system.

**Q: Do you want to allow FATSCOPY to copy ABR Archive and Application backup data sets?**

*No*<sup>x</sup> – [click here to skip ahead](#).

*Yes*<sup>x</sup> – If you haven't done so already, add **ABRARC=YES** to your COPY or SIM statement.

**Q: If Archive or Application Backup files are copied, do you want FATSCOPY to invoke FDRDSF to dump the ABR Control File as the last file on the last output tape?**

*No* – [click here to skip ahead](#).

*Yes* – Add **ARCBACKUP=DSF** to your COPY statement.

**Q: When a backup of the Archive Control File is done, the name used for the output file is normally the existing ABR Control File name with "ARCHIVE" changed to "ARCBKUP". Do you want to use a *different* file name for the Archive backup?**

*Yes* – Add **ARCB1DSN=filename** to your COPY statement.

**Q: Are the ABR modules in the linklist?**

*No* – If you haven't done so already, add a **STEPLIB** or **JOBLIB DD** statement for the ABR library.

---

<sup>x</sup> The default value for this keyword may have been changed on your system.

## General Questions: Error Handling, etc.

Finally, we will go through some general questions about how FATSCOPY handles errors and sets some limits. If you've been through this before and know that you just want to keep the FATSCOPY default values, then [click here to skip ahead](#).

**Q:** When no data sets match the selection criteria for a SELECT statements, FATSCOPY will usually issue a U0300 abend as an alert. **Do you want to suppress this abend?**

*Yes*<sup>Ⓜ</sup> – Add **SELTERR=NO** to your COPY or SIM statement. You will still get an error message notifying you that no data sets were selected. (You might use this when you expect that certain data sets might not actually exist.)

**Q:** Do you want to produce an abend dump for FATSCOPY errors which normally do not produce a dump?

*Yes*<sup>Ⓜ</sup> – Add **DUMP=YES** to your COPY or SIM statement.

**Q:** For many serious errors, FATSCOPY will produce a U0888 abend. **Do you want FATSCOPY to terminate with a return code of 12 instead?**

*Yes*<sup>Ⓜ</sup> – Add **RETCODE** to your COPY or SIM statement.

**Q:** Did you use **SELECT ALLDSN** together with **CAT=ONLY**?

*No* – [click here to skip ahead](#).

Remember that with **CAT=ONLY** the output is cataloged only if the catalog entry for the input file name exactly matches the input file actually copied (volser and file sequence). If it does not match, the file is copied but the output is not cataloged. In this case FATSCOPY issues an error message and can set a return code value of 0, 4, or 8. By default FATSCOPY will set a return code of 4<sup>Ⓜ</sup> when this mismatch occurs.

**Q:** Do you want to use a *different return code* when a mismatch is found?

*Yes*<sup>Ⓜ</sup> – Add **NOTCATRC=n** to your COPY statement, where n is 0, 4, or 8.

**You should now have a complete FATSCOPY Sim or Copy job which you can submit as a batch job.**

---

<sup>Ⓜ</sup> The default value for this keyword may have been changed on your system.

## Strategies for Copy Projects

After you have gone through this tutorial a few times and seen how FATSCOPY selects and copies datasets, you may wish to think about strategies you can use when you need to design a large tape copy project, such as a migration from one tape system to another.

Let's recall the list of questions from the beginning of the tutorial.

- Which do you have: a list of *volumes* you want to copy, or a list of *file names* (or filename patterns)?
- Do you want to *limit* the files selected by certain criteria such as expiration date, creation date, file location, file sequence number, creating job name, last-used job name, data set size, etc.?
- Where do you want the output files to go? Do you want to stack them onto as few tapes as possible? Do you want to add them to a tape which already has files on it?
- Do you want the output files to be cataloged? Should they be renamed?
- Do you want to expire the input files after they have been copied?

You should now also think about the *best way* to copy the files. For example, you might have files DEPT1.SALES.\*\* on volumes 001111 and 001112 and files DEPT1.ORDERS.\*\* on volumes 001122 and 001123 which you want to copy. Do you want to copy them in a single job, or separate jobs? To copy all of them in a single job, you might consider the following alternatives for the SELECT statements:

1. SELECT CATDSN=DEPT1.SALES.\*\*  
SELECT CATDSN=DEPT1.ORDERS.\*\*, or
2. SELECT ALLDSN,VOL=001111  
SELECT ALLDSN,VOL=001112  
SELECT ALLDSN,VOL=001122  
SELECT ALLDSN,VOL=001123

Alternative #1 may be best if you are not sure which volumes all the files are on. Alternative #2 may be best if you are sure there are no other files on those volumes. If you need to break the copying up into separate jobs, you may find one alternative to be easier to use than the other.

**There are some less-than-obvious considerations to take note of when designing your copy project. Here are some issues which you may need to think about.**

### Working with multivolume sets

SELECT ALLDSN selects all the volumes in a multivolume set, unless you use MULTIVOL=NO. Let's say you have a multivolume set 000002, 000014, and 000025. If you set up a first job with

```
SELECT ALLDSN,VOL=000001
SELECT ALLDSN,VOL=000002
...
SELECT ALLDSN,VOL=000010
```

## Strategies for Copy Projects

and a second job with

```
SELECT ALLDSN,VOL=000011
SELECT ALLDSN,VOL=000012
...
SELECT ALLDSN,VOL=000020
```

and a third job with

```
SELECT ALLDSN,VOL=000021
SELECT ALLDSN,VOL=000022
...
SELECT ALLDSN,VOL=000030
```

the multivolume set will be copied 3 times, since it will be selected by each job. There are several ways to avoid this:

- Put all the SELECT statements in 1 job. If multiple SELECT statements result in a data set being selected more than once *in a single job*, the duplicate selections are ignored, and the data set is copied only once. Or,
- Run the separate jobs with MULTIVOL=NO, then run another job with all the SELECT statements in a single job with MULTIVOL=ONLY. Or,
- (*Only if your tape management system is RMM, CA-1, or Zara*) Add EXLASTPGM=FATSCOPY to your COPY statements. Then, the second and third jobs will exclude the multivolume set since FATSCOPY will have been recorded as the last program to have used the files on those volumes after running the first job.

### How many tapes will I be copying?

When using SELECT CATDSN, you may not know how many volumes will need to be mounted to copy the data sets you have specified. When using SELECT ALLDSN, you may not know how many volumes will need to be mounted because some of the specified volumes may be part of multivolume sets. In either case, you may not know how many volumes will need to be mounted because some data sets may be bypassed because they are expired or do not meet some of the exclusion criteria you have specified.

To get a better idea of “how big” a particular FATSCOPY job will be, it is often a good idea to run a SIM with the SELECT and EXCLUDE statements for the job you are considering. The Select Report which is produced will show you which data sets have been selected and which data sets have been excluded. The Copy/Simulation Report displays a list of the input tapes which would be needed for the corresponding Copy, and an estimate of the number of output tapes which would be needed.

After reviewing these reports, you may decide that you should break the FATSCOPY job up into several smaller jobs, or that you can combine several small FATSCOPY jobs into a single larger job.

### Should I stack files together from several input tapes?

Do you want to make the most efficient use of space on your output tapes? If so, you can “stack” files from several input tapes onto as few output tapes as possible (using EXPDTGROUP=999). When the output tapes have higher capacity than the input tapes, or the input tapes have only a few small files on them, this can result in significantly fewer output tapes being used.

Stacking all the files on as few volumes as possible is not always the best strategy. Do you have files with a large range of expiration dates, and want to be able to scratch the output tapes most efficiently when each data set is no longer needed? If so, grouping by expiration date with a spread of “n” days (you specify “n” with the EXPDTGROUP= parameter) is a better choice. This will eliminate the problem of having tapes which are retained only because they have one data set on them which needs to be retained for a long time, but also contain many other data sets which are expired.

There are times when output files should not be stacked:

- Data sets which will be used at the *same time* by an application program **MUST NOT** be copied to the same volume; otherwise that program will be unable to use them unless they are copied again to separate volumes.
- When input volumes have different Owner assignments recorded by your tape management system, you may wish to maintain those assignments by copying to separate output volumes.

### Working with a large volume range

If you want to copy a large range of consecutive VOLSERS, you may find that copying all of them in one job (specified by SELECT ALLDSN,VOL=volser,NUMVOLS=range) is not convenient. The processing time window you have available at your shop may not be long enough to run a large copy job, or you may prefer to run multiple copy jobs in parallel to reduce overall processing time. FATSCOPY has a way to break up a large range of VOLSERS and generate a series of copy jobs where each one copies only a subset of the volume range. When you use the PUNCH keyword and provide the name of a PDSE or PDS (the “punch library”), FATSCOPY will generate these jobs and save them as members in the punch library. You can then run those smaller jobs as your processing time permits.

An explanation of how to use PUNCH is beyond the scope of this tutorial. Details on how to use PUNCH can be found in section 32.7 of the FATSCOPY User’s Manual, “Writing FATSCOPY Jobs to a Partitioned Data Set”.

### Using the Guided Path ISPF Panels to build an ALLDSN batch job

There is a series of ISPF panels available that uses the same approach as this Guide to step you through the procedure of generating control statements when you want to select tapes by volser (SELECT ALLDSN). On the FATSCOPY Selection Menu, option “G” invokes the Guided Path which presents panels that are displayed in a sequence based on your responses to the options on each panel. At the end of the series of panels, the full batch job is displayed, which you can submit or save to be run later.



## Strategies for Copy Projects

If you have a large number of similar copy tasks to run, differing only in the specific volsers to be copied, a useful way to use the Guided Path is to generate a job for a one or a few volsers and save that job in a library. You can edit and submit that template job over and over, changing the volser in the SELECT ALLDSN statement(s) for each submission.

## Appendix

### Sample Select Report

FATSCOPY DATA SET SELECTION REPORTS											03/04/2014 PAGE		1
DATA SET NAME	VOLSER	FSEQ	VSQ	DEVT	CRDATE	EXDATE	CREATJOB	TMSSTAT	SIZE	STATUS	REASON	PHYVOL	
JAT.EE.FATS.FILE1	CCR009	1	1		2010166	2011001	JOBORIG	USER	<1.0M	SELECT			
JAT.EE.FATS.FILE2	CCR009	2	1		2010166	2011002	JATORIG	USER	<1.0M	SELECT			
JAT.EE.FATS.FILE3	CCR009	3	1		2010166	2011003	JATORIG	USER	<1.0M	SELECT			
JAT.EE.FATS.FILE4	CCR009	4	1		2010166	2011004	JATORIG	USER	<1.0M	SELECT			
JAT.EE.FATS.FILE5	CCR009	5	1		2010166	2011005	JATORIG	USER	<1.0M	SELECT			
JAT.EE.FATS.FILE6	CCR009	6	1		2010166	2011006	JATORIG	USER	<1.0M	SELECT			
JAT.EE.FATS.FILE7	CCR009	7	1		2010166	2011007	JATORIG	USER	<1.0M	SELECT			
JAT.EE.FATS.FILE8	CCR009	8	1		2010166	2011008	JATORIG	USER	<1.0M	SELECT			
JAT.EE.FATS.FILE9	CCR009	9	1		2010166	2011009	JATORIG	USER	<1.0M	SELECT			
JAT.EE.FATS.FILE10	CCR009	10	1		2010166	2011010	JATORIG	USER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE1	CCR001	1	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE2	CCR001	2	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE3	CCR001	3	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE4	CCR001	4	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE5	CCR001	5	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE6	CCR001	6	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE7	CCR001	7	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE8	CCR001	8	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE9	CCR001	9	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
JMK1.LABELED.FATS.FILE10	CCR001	10	1		2010267	2013241	JMKREN1	MASTER	<1.0M	SELECT			
BELED.FATS.FILE11	CCR001	11	1		2010267	2011001	JMKREN1	MASTER	<1.0M	SELECT			
JMK.TSTEXPB1.DATASET	CCR045	1	1		2005319	2014334	JMKIEBDG	USER	<1.0M	SELECT			
JMK.TSTEXPB2.DATASET	CCR045	2	1		2005319	2015010	JMKIEBDG	USER	<1.0M	SELECT			
JMK.TSTEXPB3.DATASET	CCR045	3	1		2005319	2015082	JMKIEBDG	USER	<1.0M	SELECT			
JMK.T99000.CPUB.TEST	CCR045	4	1		2006166	2006166	JMKIEBDG	USER	<1.0M	BYPASS	FILESEQ		
JMK.TSTEXPB4.DATASET	CCR045	5	1		2005319	2015017	JMKIEBDG	USER	<1.0M	BYPASS	FILESEQ		

### Sample SIM Report

FATSCOPY DATA SET COPY/SIMULATION REPORT											03/04/2014 PAGE		1	
DATA SET NAME	VOLSER	FSEQ	VSEQ	EXDATE	GROUP	FUNC	O/P	EXP	O/P	VOL	I/P	TMINFO	OSEQ	RC
JAT.EE.FATS.FILE1	CCR009	1	1	2011001	1	SIM	2011001					KEEP COPY	1	0000
JAT.EE.FATS.FILE2	CCR009	2	1	2011002	1	SIM	2011002					KEEP COPY	2	0000
JAT.EE.FATS.FILE3	CCR009	3	1	2011003	1	SIM	2011003					KEEP COPY	3	0000
JAT.EE.FATS.FILE4	CCR009	4	1	2011004	1	SIM	2011004					KEEP COPY	4	0000
JAT.EE.FATS.FILE5	CCR009	5	1	2011005	1	SIM	2011005					KEEP COPY	5	0000
JAT.EE.FATS.FILE6	CCR009	6	1	2011006	1	SIM	2011006					KEEP COPY	6	0000
JAT.EE.FATS.FILE7	CCR009	7	1	2011007	1	SIM	2011007					KEEP COPY	7	0000
JAT.EE.FATS.FILE8	CCR009	8	1	2011008	1	SIM	2011008					KEEP COPY	8	0000
JAT.EE.FATS.FILE9	CCR009	9	1	2011009	1	SIM	2011009					KEEP COPY	9	0000
JAT.EE.FATS.FILE10	CCR009	10	1	2011010	1	SIM	2011010					KEEP COPY	10	0000
JMK1.LABELED.FATS.FILE1	CCR001	1	1	2013241	2	SIM	2013241					KEEP COPY	1	0000
JMK1.LABELED.FATS.FILE2	CCR001	2	1	2013241	2	SIM	2013241					KEEP COPY	2	0000
JMK1.LABELED.FATS.FILE3	CCR001	3	1	2013241	2	SIM	2013241					KEEP COPY	3	0000
JMK1.LABELED.FATS.FILE4	CCR001	4	1	2013241	2	SIM	2013241					KEEP COPY	4	0000
JMK1.LABELED.FATS.FILE5	CCR001	5	1	2013241	2	SIM	2013241					KEEP COPY	5	0000
JMK1.LABELED.FATS.FILE6	CCR001	6	1	2013241	2	SIM	2013241					KEEP COPY	6	0000
JMK1.LABELED.FATS.FILE7	CCR001	7	1	2013241	2	SIM	2013241					KEEP COPY	7	0000
JMK1.LABELED.FATS.FILE8	CCR001	8	1	2013241	2	SIM	2013241					KEEP COPY	8	0000
JMK1.LABELED.FATS.FILE9	CCR001	9	1	2013241	2	SIM	2013241					KEEP COPY	9	0000
JMK1.LABELED.FATS.FILE10	CCR001	10	1	2013241	2	SIM	2013241					KEEP COPY	10	0000
BELED.FATS.FILE11	CCR001	11	1	2011001	2	SIM	2011001					KEEP COPY	11	0000
JMK.TSTEXPB1.DATASET	CCR045	1	1	2014334	3	SIM	2014334					KEEP COPY	1	0000
JMK.TSTEXPB2.DATASET	CCR045	2	1	2015010	3	SIM	2015010					KEEP COPY	2	0000
JMK.TSTEXPB3.DATASET	CCR045	3	1	2015082	3	SIM	2015082					KEEP COPY	3	0000

\*\*\* NUMBER OF OUTPUT TAPES REQUIRED:  
 \*\*\*\* NON-COMPRESSED OUTPUT: 3  
 \*\*\*\* COMPRESSED OUTPUT (3:1 RATIO): 3

INPUT TAPES REQUIRED:  
 CCR009 CCR001 CCR045

[click here to return to the text](#)

## Appendix

### Sample Copy Report

FATSCOPY DATA SET COPY/SIMULATION REPORT											03/04/2014 PAGE		1	
DATA SET NAME	VOLSER	FSEQ	VSEQ	EXDATE	GROUP	FUNC	O/P	EXP	O/P	VOL	I/P	TMINFO	OSEQ	RC
JAT.CYCLE2.WHILCAT2	E10007	1	1	2011303	1	COPY	2011310	CCR192	KEEP	COPY			1	0000
JAT.CYCLE2.WHILCAT1	E10012	1	1	2011303	1	COPY	2011310	CCR192	KEEP	COPY			2	0000
JAT.CYCLE2.WHILCAT9	E10109	1	1	2011301	1	COPY	2011310	CCR192	KEEP	COPY			3	0000
JAT.CYCLE2.WCATGDG1.G0005V00	E10114	1	1	2011301	1	COPY	2011301	CCR192	KEEP	COPY			4	0000
JAT.CYCLE2.WCATGDG1.G0004V00	E10116	1	1	2011301	1	COPY	2011301	CCR192	KEEP	COPY			5	0000
JAT.P490597.DATASET	E10269	1	1	1999000	1	COPY	1999000	CCR192	KEEP	COPY			6	0000

[click here to return to the text](#)

## Appendix

### Sample Audit Report

FORMAT AND PRINT FATSCOPY AUDIT REPORT	VER 4.9.29	-- INNOVATION DATA PROCESSING	03/04/2014	PAGE 1										
FATAUDIT DETAIL REPORT														
JOBNAME=JATJOB1	STEPNAME=AUDIT1	RUN DATE= 2014063	RUN TIME= 161303	CPUID=CPUA										
----- 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	2011001	YES	CCR197	1	1	2011310	NO	YES	0000	
		2			24320			2		24320				
JAT.EE.FATS.FILE2	CCR009	2	1	NO	2011002	YES	CCR197	2	1	2011310	NO	YES	0000	
		2			24320			2		24320				
JAT.EE.FATS.FILE3	CCR009	3	1	NO	2011003	YES	CCR197	3	1	2011310	NO	YES	0000	
		2			24320			2		24320				

  

FORMAT AND PRINT FATSCOPY AUDIT REPORT	VER 4.9.29	-- INNOVATION DATA PROCESSING	03/04/2014	PAGE 2										
FATAUDIT DETAIL REPORT														
JOBNAME=SECONDJB	STEPNAME=AUDIT2	RUN DATE= 2014063	RUN TIME= 161335	CPUID=CPUA										
----- 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.SPAN3480.FILE	CCR005	1	1	NO	2011145	YES	CCR198	1	1	2011310	NO	YES	0000	
		40000			532000000			40000		532000000				
JAT.FBDATA.FILE	CCR163	1	1	NO	2010244	YES	CCR199	1	1	2011310	NO	YES	0000	
		50			66500			50		66500				

  

FORMAT AND PRINT FATSCOPY AUDIT REPORT	VER 4.9.29	-- INNOVATION DATA PROCESSING	03/04/2014	PAGE 3										
FATAUDIT DETAIL REPORT														
JOBNAME=JATTHIRD	STEPNAME=AUDIT3	RUN DATE= 2014063	RUN TIME= 161405	CPUID=CPUA										
----- 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.RMM.BLK65536	CCR130	1	1	NO	2011178	YES	CCR200	1	1	2011310	NO	YES	0000	
		30			1966080			30		1966080				
JAT.RMM.BLK65535	CCR130	2	1	NO	2011178	YES	CCR200	2	1	2011310	NO	YES	0000	
		20			1285000			20		1285000				

  

FORMAT AND PRINT FATSCOPY AUDIT REPORT	VER 4.9.29	-- INNOVATION DATA PROCESSING	03/04/2014	PAGE 4
FATAUDIT SUMMARY REPORT				
TOTAL INPUT VOLUMES	-	4		
TOTAL OUTPUT VOLUMES	-	4		
TOTAL BYTES READ	-	535390540		
TOTAL BYTES WRITTEN	-	535390540		
TOTAL BLOCKS READ	-	40106		
TOTAL BLOCKS WRITTEN	-	40106		
TOTAL FILES COPIED	-	7		
TOTAL FAILED COPIES	-	0		
TOTAL JOBS EXECUTED	-	3		

[click here to return to the text](#)

**Reader's Comment Form**  
**FATSCOPY How-To-Build-A-Job Guide V4.9L29**  
**INNOVATION DATA PROCESSING**

If you have comments on this product, its manual, or this tutorial, including:

- errors in the text or typographical errors;
- clarity;
- suggestions for improvement in the tutorial;
- suggestions for improvement in the product; or
- any other comments,

please complete this form and fax it to Innovation at 973-890-7147. You may also e-mail your comments to Innovation at [support@fdriinnovation.com](mailto:support@fdriinnovation.com); be sure to identify the document name in the message.

Your name: \_\_\_\_\_

Company name: \_\_\_\_\_

Mailing Address: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

E-mail address: \_\_\_\_\_

Comments: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_