ANL-8103
Mathematics
and Computers

ARGONNE NATIONAL LABORATORY
9700 South Cass Avenue
Argonne, Illinois 60439

COPYCAT - OS CATALOG UTILITY PROGRAM

by

Douglas E. Engert

Applied Mathematics Division

NOTICE

This report was prepared as an account of work sponsored by the United States Government; Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

May 1974

MASTER

TABLE OF CONTENTS

	PAGE
ABSTRACT	5
1.0 INTRODUCTION	5
2.0 USER MANUAL	6
2.1 FUNCTIONS	6
2.1.1 Redistribution	6
2.1.2 Restructuring	6
2.2.3 Master and Secondary Catalogs	7
2.1.4 Master Mode	7
2.1.5 Secondary Mode	7
2.1.6 Model Generation Data Group DSCBs	7
2.1.7 Additional CVOL Processing	7
2.1.8 Alias Processing	8
2.1.9 Multiple Inputs	8
2.2 CONTROL CARDS	8
2.2.1 CONTROL	8
2.2.2 INPUT	9
2.2.3 OUTPUT	9
2.2.4 CVOL	10
2.2.5 LEVEL	11
2.3 JOB CONTROL STATEMENTS	11
2.4 MESSAGES AND CODES	12
2.5 EXAMPLES	14
2.5.1 Copy a Catalog from One Volume to Another	14
2.5.2 Back Up a Catalog	15
2.5.3 Redistribute a Catalog	15
3.0 PROGRAM LOGIC	18

COPYCAT - OS CATALOG UTILITY PROGRAM

by

Douglas E. Engert

ABSTRACT

COPYCAT is an OS utility program designed to improve performance of OS catalog management by producing a restructured copy of a catalog in order to use the existing access methods most effectively. Multiple input and output catalogs are processed and the redistribution of the entries is under user control. Model DSCBs for generation data groups and alias entries are also processed. Catalogs on all direct access devices, including data cells, are supported. Backup copies may also be made.

1.0 INTRODUCTION

COPYCAT is an OS utility program designed to produce an efficient system wide catalog which may reside on many volumes. Substantial improvement in performance may also be obtained on a system with only a single catalog. This is accomplished in a number of ways. First, catalog entries from many different catalogs may be redistributed to equalize the load on each individual catalog. Second, each individual catalog is restructured in a way designed to minimize the I/O time required both for searching and updating. The redistribution and restructuring parameters are under the control of the user who should have some knowledge of OS catalog management.

Our installation at Argonne consists of a 370/195, used for batch processing, and a 360/75 used for time sharing. The time sharing system, TSO, shares all its online devices with the 195. Since all datasets used by TSO must be cataloged, they are cataloged on the resident volume for the 75 to allow batch jobs to access TSO datasets. This has placed an extremely heavy load on this shared catalog, which occupies about 20 cylinders on a 2314. After some study of the situation it was found that the organization of the shared catalog was one of our biggest bottlenecks. After running COPYCAT, software monitoring showed the percent of unit busy time of the system resident volume for the 75 dropped from about 80% to about 20%, the percent of enqueue wait for the catalog dropped from about 10% to about .1%, and the number of EXCPs required to logon to TSO dropped from about 120 to 70.

2.0 USER MANUAL

2.1 FUNCTIONS

Two different types of catalog systems can be processed by COPYCAT. The first is the standard OS catalog system consisting of the master catalog on the system resident volume and zero or more catalogs on other volumes. The second type of catalog system does not have a master catalog, but instead has a table built into the catalog management routines used to select the first catalog to be searched. The changes needed for this type of catalog system must be locally written.

2.1.1 Redistribution

When more than one catalog is being created, redistribution parameters are required. These parameters are supplied on the OUTPUT cards in the form of a range of names. The upper bound is defined on each OUTPUT card, and the range extends from the upper bound of the previous card to this upper bound. An EBCDIC collating sequence is used.

2.1.2 Restructuring

The main function of COPYCAT is to restructure a catalog so as to take advantage of the catalog management I/O routines. To do this three parameters may be specified for each index level. The first is the number of bytes to leave in each block for expansion. By leaving enough room, the catalog management routines will be able to add entries without having to add new blocks. Each CVOL pointer takes twenty-two (22) bytes, and single volume dataset entries take twenty-six (26) bytes. Second, the number of dummy blocks to be left after an index may be specified so the index can expand and new lower level indexes can be added in close proximity. Third, the number of blocks that must be on a track before starting the next index may be given. Since indexes may require more than one block, and an index can be searched in one I/O operation if it is on one track, this number should be set large enough so most indexes at a particular level can fit on one track.

2.1.3 Master and Secondary Catalogs

When more than one output catalog is produced, one is usually defined as the master catalog, and the others as secondary catalogs. In the case of only one output catalog, it is always a master catalog. COPYCAT must be run once to produce the master catalog in MASTER mode, and if there are secondary catalogs, once in SECONDARY mode.

2.1.4 MASTER Mode

When COPYCAT is operating in MASTER mode, only the master catalog is produced based on the OUTPUT MASTER card(s). CVOL pointers for indexes covered by the OUTPUT SECONDARY cards, if any, are also added to the master catalog.

2.1.5 SECONDARY Mode

SECONDARY mode is used to produce the secondary catalogs. The choice of which indexes are placed on which catalogs are defined on the OUTPUT cards.

2.1.6 Model Generation Data Group DSCBs

Model DSCBs for generation data groups are also handled. Since the model DSCB must reside on the volume that contains the generation index, COPYCAT will move or copy these DSCBs. The choice of whether to move or copy the DSCBs should be based on the application. When a single catalog is being rebuilt on the same physical volume, nothing need be done, since the DSCBs will still be on the volume. When a catalog is being copied to a replacement volume, such as a new release of a system, then the DSCBs should be copied. When the whole catalog system is being reorganized, then the DSCBs should be moved.

2.1.7 Additional CVOL Processing

Since every direct access volume can have a catalog, or one system may have access to another system's catalog, it may become necessary to change many CVOL pointers on one system if the other system's catalog is redistributed. The CVOL cards have been provided to accomplish this.

2.1.8 Alias Processing

Special handling of alias entries has been included in COPYCAT. In a multiple catalog system, an alias name and its true name sometimes reside in different catalogs. CVOL pointers with the alias name are placed in the master catalog during a MASTER operation and in a secondary catalog during a SECONDARY operation, pointing at the true names volume. The alias entry is then placed in the true names catalog.

2.1.9 Multiple Inputs

Thirty-two (32) catalogs may be defined as input. COPYCAT has been designed to work on a catalog system where a high level index name, other than a CVOL pointer, may only appear in one catalog. When a conflict is found, the entry from the first defined input catalog is used. An example of this would be in a two CPU system, and both system resident volume catalogs are defined as input to COPYCAT. If both catalogs have entries for SYS, then only the entries from the first defined catalog would be used.

2.2 CONTROL CARDS

The format for all control cards is:

[name] operation operands comments

The name field, if present, must begin in column one. The operation must be separated from the name and operands by at least one blank, and may start anywhere after column one. The operands must be separated by commas with no intervening blanks. A null operand must be delimited by commas unless it is the last operand. The comments are separated from the operands by at least one blank, and any card with an asterisk '*' in column one is treated as a comment card.

2.2.1 CONTROL

The CONTROL card specifies the main functions to be performed, and must be the first non-comment card.

[name] CONTROL {MASTER|SECONDARY}, {GDGMOVE|GDGCOPY},
                  {KEEP|SPLIT|UNCATLG}

MASTER|SECONDARY - COPYCAT is to be executed in the MASTER mode to produce the master catalog, or in the SECONDARY mode to produce the secondary catalogs.

GDGMOVE|GDGCOPY - Generation data group model DSCBs are to be moved or copied. If this operand is null then they are not processed.

KEEP|SPLIT|UNCATLG - The function to be performed on single name datasets. They may be kept in the master catalog (KEEP), they may be redistributed to secondary catalogs (SPLIT), or they may be uncataloged (UNCATLG).

When COPYCAT is run twice for a master catalog and secondary catalogs, these last two operands should be the same for both runs. If local modifications to the catalog management routines have been added so a master catalog is not required, COPYCAT need only be run once in SECONDARY mode, and the third CONTROL card parameter must be SPLIT.

2.2.2 INPUT

The input card defines an input catalog to be processed. There must be at least one and as many as thirty-two (32) INPUT cards. The input datasets defined by these cards are never changed by the program. They must not be the same datasets used for output. The INPUT card(s) must follow the CONTROL card.

[name] INPUT ddname, volid

ddname - The name of a DD statement that defines this input dataset.

volid - The volume name on which this catalog resides when used by the system. This volume need not be the same as that defined by ddname.

2.2.3 OUTPUT

The OUTPUT cards define the output datasets and the redistribution table. These cards must follow the input cards.

[name] OUTPUT {MASTER|SECONDARY}, ddname, volid[, maxname]

MASTER - This card defines the master catalog, or the only catalog if only one is being produced.

SECONDARY - This card defines a secondary catalog.

ddname - The name of a DD statement that defines the output dataset.

valid - The volume name on which this catalog will be used by the system. The catalog may be placed on some other volume temporarily for backup or convenience.

maxname - The highest name in EBCDIC collating sequence to which this OUTPUT card applies. This maxname must be greater than the maxname on the previous OUTPUT card, and on the last OUTPUT card it must be null.

Each OUTPUT card defines a range of index names that are to be placed in its catalog. This range extends from the previous maxname up to and including the current maxname. The same ddname and valid pair may appear on more than one card. This allows more than one range of index names to be placed on one catalog. All MASTER cards must have the same ddname and valid since there can only be one master catalog. The last OUTPUT card must have a null maxname.

2.2.4 CVOL

The CVOL card may be used to alter control volume pointer entries that point to catalogs outside the current system.

[name] CVOL [valid],newvol,[maxname],ddname

valid - The volume name that is being changed.

newvol - The new volume name for the CVOL entry.

maxname - The highest name in the range of index names covered by this CVOL card. All the CVOL pointers that point at this volume, whose name is greater than the previous maxname and less than or equal to this maxname, will be changed to point at newvol. The last CVOL card in a set must have a null value for maxname, which is the highest in the collating sequence.

ddname - The name of a DD statement used for device type information. This DD statement must define a unit of the same type as newvol.

Many sets of CVOL cards may be defined. Each set must have valid specified on the first card, and maxname must be null on the last. CVOL cards must follow the OUTPUT cards.

2.2.5 LEVEL

The LEVEL card is used to specify the restructuring information. The greatest performance improvements derived from running COPYCAT depend on properly choosing these parameters. Up to twenty-two LEVEL cards may be defined.

[name] LEVEL level,bytes,dummies,blocks

level - The index level between one and twenty-two to which this card applies.

bytes - The number of bytes to leave in each block for future expansion, between 0 and 200.

dummies - The number of dummy blocks to leave after this index for future expansion.

blocks - The number of blocks that must be on the current output track before starting this index.

The LEVEL cards must follow the CVOL cards, if any. A default value of zero is applied for any null or missing values.

2.3 JOB CONTROL STATEMENTS

A JOB card and an EXEC card are required. A SYSPRINT DD statement is required for a listing of the input, informational messages and error messages. The DCB parameters are fixed at RECFM=PBA,LRECL=121,BLKSIZE=121. The control cards are read from the SYSIN file which has fixed DCB parameters of RECFM=PB,LRECL=80,BLKSIZE=80. Additional DD statements are required as specified by the INPUT, OUTPUT and CVOL cards. The datasets defined by these DD statements need not be named SYSCTLG, but they must be on a direct access volume. If a catalog is created on a volume other than the one on which it will be used by the system, the temporary volume should be of the same device type as the final volume.

2.4 MESSAGES AND CODES

After a successful run of COPICAT certain statistics about each output dataset are listed. The last entry is the dataset and the TTR of the last used block is listed in hexadecimal. The number of control volume blocks, dummy blocks, blocks at levels one and two and blocks cataloged are listed in decimal.

The following is a list of error messages and completion codes. The maximum completion code found in the output of the program is returned to OS.

ERR01 ERROR FOUND IN PARAMETER AT '\$'

An error was found in a parameter or a required parameter was missing. The previous line will contain a '\$' under the error.

(Completion code = 8)

ERR02 ERROR PROCESSING COMMAND=command

A problem was found with a DS statement. Either it was not found, could not be opened, or was not on a direct access volume.

(Completion code = 9)

ERR03 UNEXPECTED END OF FILE IN SYM

More control volume blocks were expected, but were not found.

(Completion code = 10)

ERR04 NAME=aliasname ALIAS=aliasname

While processing the entry, the true name for the alias could not be found.

(Completion code = 11)

ERR05 ALIAS COUNT FOR NAME=truename

The alias count for the named index is not equal to the number of aliases for the name.

(Completion code = 12)

ERR06 TOO MANY INPUT DATA SETS

More than 32 input datasets were specified.

(Completion code = 13)

ERR07 INDEX=index FOUND ON VOLS=(x,y) THE ENTRY ON x WILL BE USED

The same high level index was found on two different catalogs. The one on the x volume will be used.
(Completion code = 4)

ERR08 NO INPUT CARDS FOUND

No INPUT cards were found, or they did not follow the CONTROL card.
(Completion code = 8)

ERR09 function FAILED WITH (code) ON volume FOR GDG MODEL DSCB=name

While moving or copying model DSCBs, an OBTAIN, ALLOCATE or SCRATCH operation failed with the return code of 'code' (decimal).
(Completion code = 4)

ERR10 MODEL DSCB FOR GDG DOES NOT HAVE ZERO EXTENTS DSN=name

The named model DSCB is not copied because it does not have zero extents.
(Completion code = 4)

ERR11 MODEL DSCB FOR GDG EXISTS ON OUTPUT VOLUME WITH DIFFERENT ATTRIBUTES FOR DSN=name

While moving or copying model DSCBs a DSCB with the same name was found on the output volume, but the attributes defined by the twelve bytes in the DSCB at offset 82 (decimal) are not the same. The DSCB is not moved or copied.
(Completion code = 4)

ERR12 ABEND - nnn - OCCURED. SEE SYSMSG. CURRENT OR LAST NAME IS

ERR12 ABEND - nnn - OCCURED. SEE SYSMSG. CURRENT OR LAST NAME IS
A 037, 837 or 213 abend occurred. The last index structure being processed is given.
(Completion code = 16)

ERR13 I/O ERROR - synad message

ERR13 I/O ERROR CURRENT OR LAST NAME IS name

An I/O error occurred. The standard message produced by a SYNAD macro is given, along with the last index structure processed.
(Completion code = 16)

ERR14 NO OUTPUT MASTER CARD WAS FOUND

When in master mode, at least one OUTPUT MASTER card must be defined.

(Completion Code = 8)

The completion codes are as follows:

- 0 - No errors.
- 4 - All catalogs are useable, but duplicate indexes have been left out or model DSCBs have not been moved or copied.
- 8 - Errors in processing control cards. The output datasets have not been changed.
- 12 - All catalogs are useable, but problems exist with alias entries.
- 16 - Output catalogs have been partly built, and are not useable.

2.5 EXAMPLES**2.5.1 COPY A CATALOG FROM ONE VOLUME TO ANOTHER**

This first example illustrates the most common use of COPYCAT. A catalog on one volume is to be copied to another volume, such as when a new system is being installed. This job would be run under the old system to copy the catalog on SYSRES to the new system pack, NEWRES, which will later be relabeled SYSRES.

```
//EX1    JOB    XXI
//        EXEC  PGM=COPYCAT,REGION=60K-
//SYSPRINT DD  SYSOUT=A
//OLDBRES DD   DSN=SYSCTLG,VOL=SER=SYSRES,
//              DISP=OLD,UNIT=DISK
//NEWRES DD    DSN=SYSCTLG,VOL=SER=NEWRES,
//              DISP=OLD,UNIT=DISK
//SYSIN  DD    *
*          EXAMPLE ONE
          CONTROL MASTER,GDGCOPY,KEEP
          INPUT OLDBRES,SYSRES
          OUTPUT MASTER,NEWRES,SYSRES
          LEVEL 1,26,100
          LEVEL 2,52,5,5
```

In this example model DSCBs for generation data groups are copied. 100 dummy blocks are left after the highest level index, with 26 bytes left in each block for future expansion. 52 bytes are left in each second level index block, 5 blocks are left after each index, and a second level index will not be started on a track within 5 blocks of the end.

2.5.2 Back Up a Catalog

This example demonstrates the use of COPYCAT for backing up a catalog. Note that the backup dataset does not need to be called SYSCTLG, thus many catalogs could be backed up on the same volume.

```
//EX2    JOB    XXX,REGION=60K
//        EXEC   PGM=COPYCAT.
//SYSPRINT DD  SYSOUT=A
//CATALOG DD   DSNAME=SYSCTLG,VOL=SER=PACK01,
//              UNIT=DISK,DISP=OLD
//BACKUP DD    DSNAME=CAT.PACK01,VOL=SER=BACKUP,
//              UNIT=DISK,SPACE=(CYL,(3,2)),DISP=(NEW,KEEP)
//SYSIN  DD    *
*           EXAMPLE 2
           CONTROL MASTER,GDGCOPY,KEEP
           INPUT CATALOG,PACK01
           OUTPUT MASTER,BACKUP,PACK01
```

2.5.3 Redistribute a Catalog

In this example a single catalog is to be split up into four separate catalogs. The first step produces the secondary catalogs, and the second step produces the master catalog. The last step is executed only if the first two steps ran properly, and it renames the dataset on RES075.

```
//EX3    JOB    XXX
//STEP1  EXEC   PGM=COPYCAT,REGION=60K
//SYSPRINT DD  SYSOUT=A
//INPUT  DD    VOL=SER=RES075,UNIT=DISK,DISP=OLD,
//              DSNAME=SYSCTLG
//OLDRES DD    VOL=SER=RES075,UNIT=DISK,
//              DSNAME=NEWCAT,DISP=(NEW,KEEP),
//              SPACE=(CYL,(5,2))
//P87    DD     UNIT=DISK,VOL=SER=PACK87,
//              DSNAME=SYSCTLG,DISP=(NEW,KEEP),
```

```

//          SPACE= (CYL, (5, 2))
//P88      DD      UNIT=DISK, VOL=SER=PACK88,
//          DSNAME=SYSCTLG, DISP= (NEW, KEEP) ,
//          SPACE= (CYL, (5, 2))
//P89      DD      UNIT=DISK, VOL=SER=PACK89,
//          DSNAME=SYSCTLG, DISP= (NEW, KEEP) ,
//          SPACE= (CYL, (5, 2))
//SYSIN    DD      *
*          EXAMPLE 3
*          PRODUCE THE SECONDARY CATALOGS
CONTROL SECONDARY, GDGMOVE, KEEP
INPUT INPUT, RES075
OUTPUT SECONDARY, P87, PACK87, B9999999
OUTPUT SECONDARY, P88, PACK88, H9999999
OUTPUT MASTER, OLDRES, RES075, K9999999
OUTPUT SECONDARY, P89, PACK89, R9999999
OUTPUT MASTER, OLDRES, RES075, S9999999
OUTPUT SECONDARY, P88, PACK88
LEVEL 1, 26, 100
LEVEL 2, 54, 5, 5
LEVEL 3, , 2, 2
/*
//STEP2    EXEC    PGM=COPICAT, REGION=60K
//SYSPRINT DD      SYSOUT=A
//INPUT    DD      VOL=SER=RES075, UNIT=DISK, DISP=OLD,
//          DSNAME=SYSCTLG
//OLDRES   DD      VOL=SER=RES075, UNIT=DISK,
//          DSNAME=NEWCAT, DISP=OLD
//P87      DD      UNIT=DISK, VOL=SER=PACK87,
//          DSNAME=SYSCTLG, DISP=OLD
//P88      DD      UNIT=DISK, VOL=SER=PACK88,
//          DSNAME=SYSCTLG, DISP=OLD
//P89      DD      UNIT=DISK, VOL=SER=PACK89,
//          DSNAME=SYSCTLG, DISP=OLD
//SYSIN    DD      *
*          EXAMPLE 3
*          PRODUCE THE MASTER CATALOG
CONTROL MASTER, GDGMOVE, KEEP
INPUT INPUT, RES075
OUTPUT SECONDARY, P87, PACK87, B9999999
OUTPUT SECONDARY, P88, PACK88, H9999999
OUTPUT MASTER, OLDRES, RES075, K9999999
OUTPUT SECONDARY, P89, PACK89, R9999999
OUTPUT MASTER, OLDRES, RES075, S9999999
OUTPUT SECONDARY, P88, PACK88
LEVEL 1, 26, 50
LEVEL 2, 54, 10, 10
/*
//STEP3    EXEC    PGM=IEHPROGM, COND=(5, LT)
//SYSPRINT DD      SYSOUT=A
//OLDRES   DD      UNIT=DISK, VOL=SER=RES075, DISP=OLD
//SYSIN    DD      *

```

```
RENAME VOL=2314=RES075,DSNAME=SYSCTLG,  
      NEWNAME=OLDCATLG  
RENAME VOL=2314=RES075,DSNAME=NEWCAT,  
      NEWNAME=SYSCTLG
```

All indexes up to C will be on PACK87, indexes from C to I and from T upwards will be on PACK88, and from L to S will be on PACK89. CVOL pointers for these indexes will be placed on RES075 along with all the other indexes. Step 3 will be executed if there are no severe errors.

3.0 PROGRAM LOGIC

COPYCAT is written entirely in assembler language. OSAM is used for SYSPRINT and SYSIN, while BSAM in the load mode is used for the output datasets. BDAM is used for the input and to access previously written output blocks.

The heart of the program consists of a recursive routine called RR, which is called for each level. The TTR of the first block of the level is passed to RR. After an index is written with the old TTRs, it is reread and for each entry with a lower level, RR is recursively called. Upon return, the new TTR is placed in the entry and the block is rewritten.

When processing the highest level index, the additional information of which input dataset this entry came from must also be saved. This is done by changing the count field of the index. Only three types of entries need this information saved: the index pointer entry, the volume control block pointer entry, and the generation index pointer entry. By making the last three bits of the volume control block pointer entry '110', only the entries with a last bit of '0' contain additional information. Two bits are used to distinguish between the three types of entries, and the other five bits in the length field are then used to locate the proper input dataset.

When processing model DSCBs, the obtain, allocate and scratch SVCs are used. Before running COPYCAT, local restrictions on allocating and scratching datasets should be considered. These SVCs are only used in the routine called GDSPROC.

Two other important routines are INR and PUTC. INR finds the next name in the highest level indexes to be processed from among all the inputs. It is called by RR and it handles the problem of duplicate index names. PUTC is also called by RR and it takes an index and adds it to the next block for output. PUTC processes the parameters of the LEVEL cards.