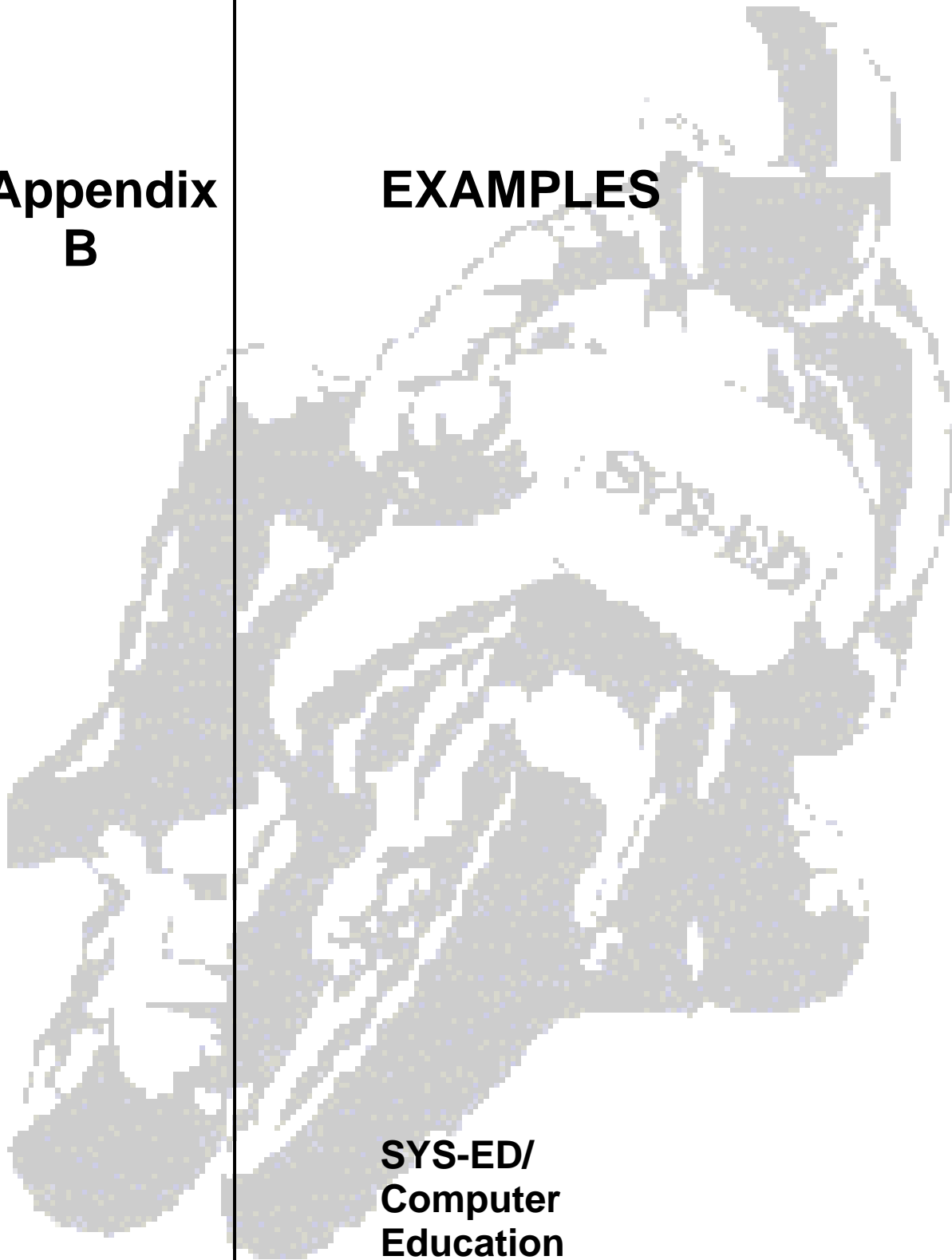


**Appendix  
B**

**EXAMPLES**



**SYS-ED/  
Computer  
Education  
Techniques, Inc.**

---

## 1 IEBGENER

---

### 1.1 Printing a Sequential Data Set

```
//PRINT JOB
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY //SYSUT1 DD DSN=D80.DATA,DISP=SHR
//SYSUT2 DD SYSOUT=A
```

---

### 1.2 Create a Partitioned Data Set from Sequential Input

```
//TAPEDISK JOB
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSN=INSET,UNIT=tape,LABEL=(,SL),
//  DISP=(OLD,KEEP),VOLUME=SER=001234
//SYSUT2 DD DSN=NEWSET,UNIT=disk,DISP=(,KEEP),
//  VOLUME=SER=11112,SPACE=(TRK,(10,5,5)),
//  DCB=(RECFM=EB,LRECL=80,BLKSIZE=2000)
//SYSIN DD *
  GENERATE MAXNAME=3,MAXGPS=2
  MEMBER NAME=MEMBER1
  GROUP1 RECORD IDENT=(8,'FIRSTMEM',1)
  MEMBER NAME=MEMBER2
  GROUP2 RECORD IDENT=(8,'SECNDMEM',1)
  MEMBER NAME=MEMBER3
//
```

GENERATE indicates that three member names are included in subsequent MEMBER statements and that the IDENT parameter appears twice in subsequent RECORD statements.

- The first MEMBER statement assigns a member name (MEMBER1) to the first member.
- The first RECORD statement (GROUPI) identifies the last record to be placed in the first member. The name of this record (FIRSTMEM) appears in the first eight positions of the input record.
- The remaining MEMBER and RECORD statements define the second and third members.
- There is no RECORD statement associated with the third MEMBER statement, the remainder of the input file will be loaded as the third member.

### 1.3 Convert Sequential Input into Partitioned Members

```
// DISKTODK JOB
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DO SYSOUTA
//SYSUTI DD DSN=INSET, UNIT=disk, DISP=(OLD,KEEP),
//          VOLUME=SER=1 111 12,LABEL=(,SUL)
//SYSUT2 DO DSN=EXISTSET, UNIT=disk, DISP=(MOD,KEEP),
//          VOLUME=SER=1 11113 b/SYSIN DD .
      GENERATE MAXNAME=3,MAXGPS=1
      EXITS INHDR=ROUTI ,INTLR=ROUT2
      MEMBER NAME=(MEMX,ALIASX)
GROUP1 RECORD IDENT=(8, 'FIRSTMEM', 1)
      MEMBER NAME=MEMY
/*
```

GENERATE indicates that a maximum of three names and aliases are included in subsequent MEMBER statements and that one IDENT parameter appears in a subsequent RECORD statement.

EXITS defines the user routines that are to process user labels.

- The first MEMBER statement assigns a member name (MEMX) and an alias (ALIASX) to the first member.
- The RECORD statement identifies the last record to be placed in the first member. The name of this record (FIRSTMEM) appears in the first eight positions of the input record.
- The second MEMBER statement assigns a member name (MEMY) to the second member.
- The remainder of the input data set is included in this member.

---

## 2 IEBCOPY

---

### 2.1 Copying an Entire Data Set

A partitioned data set (DATASET5) is copied from one disk volume to another.

```
//COPY JOB
//JOBSTEP EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUTI 00 DSNAME=DATASET5,UNIT=disk,VOL=SER=1 11113
//      DISP=SHR
//SYSUT2 DD DSNAME=DATASET4,UNIT=disk,VOLSERI 11112,
//      DISP=(NEW,KEEP),SPACE=(TRK,(5,1 ,2))
```

---

### 2.2 Merging Four Data Sets

Members are copied from three input partitioned data sets (DATASET1, DATASET5, and DATASET6) to an existing output partitioned data set (DATASET2).

The sequence in which the control statements occur controls the manner and sequence in which partitioned data sets are processed.

```
IICOPY JOB
/IJOBSTEP EXEC PGM=IEBCOPY f/SYSPRINT DD SYSOUT=A
/fINI  DD DSNAME=DATASET1 ,UNIT=disk,VOL=SER1 11112,
/f      DISP=SHR
/bIN5  DD DSNAME=DATASET5,UNIT=disk,VOLSER1 11114,
//      DISP=OLO
bbOUT2 DD DSNAME=DATASET2,UNIT=disk,VOL=SER=1 11115,
bb      DISP=(OLD,KEEP)
//IN6  DD DSNAME=DATASET6,UNIT=disk,VOL=SER=1 11117,
/b      DISP=(OLD,DELETE)
/bSYSUT3 DD UNIT=SYSOA,SPACE=(TRK,(1)) /ISYSIN DD .
COPYOPER COPY OUTOD=OUT2
        INDD=IN1
        INDD=IN6
        INDD=IN5
```

---

### 2.3 Copying and Replacing Selected Members of a Data Set

Two members (A and B) are selected from two input partitioned data sets (DATASET5 and DATASET6) and copied to an existing output partitioned data set (DATASET1).

Member B replaces an identically named member that already exists on the output data set.

```
//COPY JOB
//JOBSTEP EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//OUT1 DD DSNAME=DATASET1 ,UNIT=disk,VOL=SER1 11112,
// DISP=(OLD,KEEP)
// 1N6 DD DSNAME=DATASET6, UNIT=disk,VOL=SER1 11115,
// DISP=OLD
// 1N5 DD DSNAME=DATASET5, UNIT=disk,VOL=SER1 11116,
// DISP=(OLD,KEEP)
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(1))
//SYSUT4 DD UNIT=SYSDA,SPACE=(TRK,(1))
//SYSIN DD
COPYOPER COPY OUTDD=OUT1
        INDD=1N5,1N6
        SELECT MEMBER=((B,,R),A)
/*
```

---

### 2.4 Unloading and Compressing a Data Set

A partitioned data set is unloaded to a tape volume to create a backup copy of the data set. If this step is successful, the partitioned data set is to be compressed in place.

```
//SAVE JOB
//STEP1 EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=PARTPDS, UNIT=disk,VOL=SER=PCPOOL,
// DISP=OLD
//SYSUT2 DD DSNAME=SAVDATA, UN IT=tape,VOL=SER=TAPEO3,
// DISP=(NEW,KEEP),LABEL=(,SL)
//SYSUT3 DD DSNAME=TEMP1 .UNIT=disk,VOL=SER=1 11111,
// DISP=(NEW,DELETE),SPACE=(80,(60,45)) //SYSIN DD DUMMY
//STEP2 EXEC PGM=IEBCOPY,COND=(0,NE),PARM=SIZE=500K'
//SYSPRINT DD //SYSOUT=A
//COMPDS DD DSNAME=PARTPDS,UNIT=disk,DISP=OLD,
// VOL=SER=PCPOOL
//SYSUT3 DD DSNAME=TEMPA, UNIT=disk,VOL=SER=1 11111,
// DISP=(NEW,DELETE),SPACE=(80,(60,45)) //SYSIN DD
        COPY OUTDD=COMPDS, INDD=COMPDS
/*
```

---

## 2.5 Multiple Copy Operations with One Output Data Set

Members are selected, excluded, and copied from input partitioned data sets onto an output partitioned data set.

```
//COPY JOB ...
//JOBSTEP EXEC PGM=IEBCOPY
//ISYSRINT DO SYSOUTH A
//INOUTA DD DSN=DATASET A, UNIT=disk, VOL=SER=1 11113,
//      DISP=OLD
//INB DD DSN=DATASET B, UNIT=disk, VOL=SER=1 11115,
//      DISP=(OLD,KEEP)
//INC DD DSN=DATASET C, UNIT=disk, VOL=SER=1 11114,
//      DISP=(OLD,KEEP)
//IND DD DSN=DATASET D, UNIT=disk, VOL=SER=1 11116,
//      DISP=OLD
//INE DD DSN=DATASET E, UNIT=disk, VOL=SER=1 11117,
//      DISP=OLD
//OUTX DD DSN=DATASET X, UNIT=disk, VOL=SER=1 11112,
//      DISP=(NEW,KEEP),SPACE=(TRK,(3,1,2))
//SYSUT3 DD UNIT=SYSDA,SPACE=(TRK,(1))
//SYSIN DD *
COPERST1 COPY O=OUTX,I=INOUTA
        COPY OUTDD=INOUTA,INDD=INOUTA
        INDD=INB
        COPY OUTDD=INOUTA
        INDD=IND
        EXCLUDE MEMBER=MM
        INDD=INC
        SELECT MEMBER=((ML,MD,R))
        INDD=INE
/*
```

---

## 2.6 Loading a Data Set

A sequential data set created by an IEBCOPY unload operation is loaded.

```
//LOAD JOB
//STEP1 EXEC PGM=IEBCOPY
//ISYSRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=UNLOADSET, UNIT=tape,LABEL=(,SL),
//      VOL=SER=TAPE01 ,DISP=OLD
//SYSUT2 DD DSNAME=DATASET4, UNIT=disk,VOL=SER=2222222,
//      DISP=(NEW,KEEP),SPACE=(CYL,(1 0,5,10))
//SYSUT3 DD OSN=TEMP1 ,UNIT=disk,VOL=SER=1 11111,
//      DISP=(NEW, DELETE),SPACE~(80,(1 5,1))
//SYSIN DD DUMMY
/*
```

---

### 3 IEBCOMPR

---

#### 3.1 Comparing Data Sets that Reside on Tape

Two sequential data sets that reside on 9-track tape volumes are being compared.

```
//TAPETAPE JOB
// EXEC PGM=IEBCOMPR b/SYSPRINT DD SYSOUT=A
//SYSUT1 OD DSNAME=SET1 UNIT=tape, LABEL=(, NL),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=2000),
// DISP=(OLO, KEEP),VOLUME=SER=001 234
//SYSUT2 DD DSNAME=SET2, UNIT=tape,LABEL=(,NL),
// OCB=(RECFM=FB, LRECL=80, BLKSIZE=1 040),
// DISP~(OLD,KEEP),VOLUME=SER=001 235 b/SYSIN DO DUMMY
/*
```

---

#### 3.2 Comparing Sequential Data Sets

Two sequential data sets that reside on tape volumes are being compared.

```
//TAPETAPE JOB
// EXEC PGM=IEBCOMPR
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DSNAME=SET1 LABEL=(2,SUL),DISP=(OLD, KEEP),
// VOL=SER001 234, DCB=(DEN=2,RECFM=FB, LRECL=80,
// BLKSIZE=2000,TRTCH=C), UNIT=tape
//SYSUT2 OD DSNAME=SET2, LABEL=(,SUL), DISP=(OLD, KEEP),
// VOL=SER=001235,DCB=(DEN=2, RECFM=FB,LRECL=80,
// BLKSIZE2000,TRTCHC), UNIT=tape
//SYSIN DD *
// COMPARE TYPORG=PS
// LABELS DATA=ONLY
/*
```

SYSUT1 DD defines an input data set, SET1, which resides on a labeled, 7-track tape volume. The blocked data set was originally written at a density of 800 bits per inch (DEN=2) with the data converter on (TRTCH=C).

COMPARE TYPORGPS specifies that the input data sets are sequentially organized.

LABELS DATA=ONLY specifies that user header labels are to be treated as data and compared.

All other labels on the tape are ignored.

---

### 3.3 Comparing Two Partitioned Data Sets

Two partitioned data sets are being compared.

```
//DISKDISK JOB ...  
//STEP1 EXEC PGM=IEBCOMPR  
//SYSPRINT DO SYSOUT=A  
//SYSUT1 DD DSNAME=PDSSET1 UNIT=disk, DISP=SHR,  
//      DCB=(RECFM=FB, LRECL=80,BLKSIZE=2000),  
//      VOLUME=SER=111112  
//SYSUT2 DD DSNAME=PDSSET2,UNIT=disk,DISP~SHR,  
//      DCB=(RECFM=FB, LRECL=80,BLKSIZE=2000),  
//      VOLUME=SER=111113  
//SYSIN DD *  
        COMPARE TYPORG=PO  
/*
```

---

## 4 IEBTPCH

---

### 4.1 Printing a Partitioned Data Set

---

```
PRINT TYPORG=P0, MAXNAMS=1, MAXFLDS=1  
MEMBER NAME=UTILUPD8  
RECORD FIELD=(80)
```

### 4.2 Printing a Sequential Data Set According to Default Format

---

```
//PRINT JOB ...  
//STEPI EXEC PGM=IEBTPCH  
//SYSPRINT DD SYSOUT=A  
//SYSUTI DD DSNAME=INSET,UNIT=tape  
// LABEL=(,NL),VOLUMESER001234,  
// DISP=(OLD, KEEP), DCB=(RECFM=U, BLKSIZE=2000)  
// SYSUT2 DD SYSOUT=A  
// SYSIN DD *  
PRINT TOTCONV=XE  
TITLE ITEM=('PRINT SEQ DATA SET WITH CONV TO HEX',IO)  
/*
```

PRINT begins the print operation and specifies conversion from alphanumeric to hexadecimal representation.

TITLE specifies a title to be placed beginning in column 10 of the printed output. The title is not converted to hexadecimal.

---

**5 IEBDG**

---

**5.1 Placing Binary Zeros in Records Copied from Sequential Data Set**

---

```
//CLEAROUT JOB...
//STEP1 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=A
//SEQIN DD DSNAME=INSET,UNIT=tape, DISP=(OLD, KEEP),
//   DCB=(RECFM=FB, LRECL=80,BLKSIZE=800)
//   LABEL=(,NL),VOLUME=5ER22222
//   SEQOUT DD DSNAME=OUTSET,UNIT=tape, DISP(,KEEP),
//   DCB=(RECFM=FB, LRECL=80, BLKSIZE=800),
//   VOLUME=SER=222333,LABEL=(,NL)
//   ISYSIN DD,
DSD OUTPUT=(SEQOUT), INPUT=(SEQIN)
ED NAME=FIELD1, LENGTH=10, STARTLOC=20
FD NAME=FIELD2, LENGTH=10, STARTLOC=50
CREATE QUANTITY~100, INPUT=SEQIN, NAME=(FIELD1, FIELD2)
END
/*
```

DSD marks the beginning of a set of utility control statements and refers to the DD statements defining the input and output data sets.

The first and second FD statements create two 10-byte fields (FIELD1 and FIELD2). Because no pattern is specified for these fields, each field contains the default fill of binary zeros. The fields are to begin in the 20th and 50th bytes of each output record.

CREATE constructs 100 output records in which the contents of previously defined fields (FIELD1, FIELD2) are placed in their respective starting locations in each of the output records. Input records from data set INSET are used as the basis of the output records.

## 5.2 Ripple 10-byte Alphabetic Pattern

```
//RIPPLE JOB.
//STEP1 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=A
//SEQIN DD DSN=INSET, UNIT=tape, VOL=SER=222222,
//      DCB=(RECFM=FB, LRECL=80, BLKSIZE=800),
DISP=(OLD, KEEP)
//SEQOUT DD DSN=OUTSET, UNIT=disk, VOLUME=SER=111111
//      DCB=(RECFM=FB, LRECL=80, BLKSIZE=800),
//      DISP=(, KEEP), SPACE=(TRK, (10, 10))
//SYSIN DD *
      DSD OUTPUT=(SEQOUT), INPUT=(SEQIN)
      FD NAME=FIELD1 ,LENGTH=10,FORMAT=AL,ACTION=RP,STARTLOC=1
      CREATE QUANTITY=100, INPUT=SEQIN, NAME=FIELD1
      END
/*
```

The FD statement creates a 10-byte field in which the pattern ABCDEFGHIJ is initially placed. The data is rippled after each output record is written.

CREATE constructs 100 output records in which the contents of a previously defined field (FIELD1) are included. The CREATE statement uses input records from data set INSET as the basis of the output records.

## 5.3 Output Records at Job Step Completion

72

```
//UTLYONLY JOB ...
//STEP1 EXEC PGM=IEBDG
//SYSPRINT DD SYSOUT=A
//SEQOUT DD DSN=OUTSET, UNIT=disk, DISP=(, KEEP),
//      DCB=(RECFM=FB, LRECL=80, BLKSIZE=800),
//      SPACE=(TRK, (10, 10)), VOLUME=SER=111111
//SYSIN DD DATA
      DSD OUTPUT=(SEQOUT)
      FD NAME=FIELD1, LENGTH=30, STARTLOC=1, FORMAT=AL, ACTION=TL
      FD NAME=FIELD2, LENGTH=30, STARTLOC=31, FORMAT=AL, ACTION=TR
      FD NAME=FIELD3, LENGTH=10, STARTLOC=71, PICTURE=10,
      P'1234567890', INDEX=1
      CREATE QUANTITY=100, NAME=(FIELD1, FIELD2, FIELD3), EILL=X'FF'
      END
/*
```

DSD marks the beginning of a set of utility control statements and refers to the DO statement defining the output data set.

FD defines the contents of three fields to be used in the construction of output records.

- The first field contains 30 bytes of alphabetic data to be truncated left after each output record is written.
- The second field contains 30 bytes of alphabetic data to be truncated right after each output record is written.
- The third field is a 10-byte field containing a packed decimal number (1234567890) to be increased by one after each record is written.

CREATE constructs 100 output records in which the contents of previously defined fields (FIELD1, FIELD2, and FIELD3) are included.

After each record has been written, FIELD1 and FIELD2 are restored to full width.

---

## 6 IEHLIST

---

### 6.1 List CVOL Entries

---

```
//CATLIST JOB
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD2 DD UNIT=disk,VOLUMESER1 11111 ,DISP=OLD
//SYSIN DD *
LISTCTLG VOL=disk=111111
/*
```

LISTCTLG defines the source volume and specifies the list operation.

---

### 6.2 List Selected CVOL Entries

---

```
//CATLIST JOB ...
//STEP1 EXEC PGM=IEHLIST
//SYSPRINT DD SYSOUT=A
//DD1 DD UNITdiskB,VOLUMESER1 11111,DISP=OLD
///DD2 DD UNIT=(diskA,,DEFER),DISP=OLD,
// VOLUME=(PRIVATE,,SER=(222222))
//SYSIN DD *
LISTCTLG
LISTCTLG VOL=diskA=333333
LISTCTLG VOL=diskA=444444
LISTCTLG VOL=diskA=555555, NODE=A.B.C
/*
```

The first LISTCTLG statement indicates that the CVOL residing on the system residence volume is to be listed.

The second and third LISTCTLG statements identify two diskA disk volumes containing CVOLs to be listed.

The fourth LISTCTLG statement identifies a diskA volume containing an CVOL that is to be partially listed.

All data set entries whose beginning qualifiers are "A.B.C" are listed.

---

### 6.3 List Partitioned Directories Using DUMP and FORMAT

```
//LISTPDIR JOB ...  
//STEP1 EXEC PGM=IEHLIST  
//SYSPRINT DD SYSOUT=A  
//DD1 DD UNIT=diskB,VOLUME=SER=111111,DISP=OLD  
//DD2 DO UNIT=diskA,VOLUME=SER=222222,DISP=OLD  
//SYSIN DD *  
    LISTPDS DSNAME=PDSE1  
    LISTPDS DSNAME(PART1 ,PART2),VOLdiskA=222222,FORMAT  
/*
```

The first LISTPDS statement indicates that the PDSE directory belonging to data set PDSE1 is to be listed. The listing is in unedited (dump) format. This data set resides on the system residence volume.

The second LISTPOS statement indicates that partitioned data set directories belonging to data sets PART1 and PART2 are to be listed. The listing is in unedited (dump) format.

These data sets exist on a disk volume (222222).

---

## 7 IEHPROGM

---

### 7.1 Scratch Temporary System Data Sets

```
//SCRVTOC JOB
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD2 DD UNIT=disk,VOLUME=SER222222, DISP~OLD
//SYSIN DD .
    SCRATCH VTOC,VOL=disk=222222,SYS
/*
```

The SCRATCH statement, with SYS specified, indicates that all temporary system data sets whose expiration dates have expired are scratched from the specified volume.

---

### 7.2 Scratch and Uncatalog Two Data Sets

```
//SCRDSETS JOB ...
///STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1 DD UNIT=disk,VOLUME=SER=111111,DISP=OLD
//DD2 DD UNIT=disk,DISP=OLD,VOLUMESER222222
//SYSIN DD *
    SCRATCH DSNAMESET1,VOL=disk=222222
    UNCATLG DSNAME=SET1
    SCRATCH DSNAME=A.B.C.D.E,VOL=disk=222222
    UNCATLG DSNAME=A.B.C.D.E
/*
```

The first SCRATCH statement specifies that SET1, which resides on volume 222222, is scratched.

The first UNCATLG statement specifies that SET1 is uncataloged.

The second SCRATCH statement specifies that A. B.C.D.E, which resides on volume 222222, is scratched.

The second UNCATLG statement specifies that A. B.C.D.E is uncataloged.

---

### 7.3 Uncatalog Three Data Sets

```
//DLTSTRUC JOB ...
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1 DO UNIT=disk, VOLUMESER111111, DISP=OLD
//SYSIN DD *
  UNCATLG DSNAME=A.B.C.D.E.F.SET1
  UNCATLG DSNAME=A.B.C.G.H.SET2
  UNCATLG DSNAME=A.B.I.J.KSET3 1*
```

---

### 7.4 Rename a Partitioned Data Set Member

```
//REN JOB
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//DD1 DD VOLSER222222, DISPOLD, UNIT=disk
//SYSIN DD *
  RENAME VOL=disk222222,DSNAMEDATASET, NEWNAME=BC, MEMBER=ABC
/*
```

RENAME specifies that member ABC in the partitioned data set DATASET, which resides on a disk volume, is renamed BC.

## 7.5 Create Model DSCB and Build Generation Data Group Index

```
//BLDINDX JOB...
//STEP1 EXEC PGM=IEHPROGM
//SYSPRINT DD SYSOUT=A
//BLDDSCB DD DSNAME=A.B.C,DISP=(,KEEP),SPACE=(TRK,(0)),
//      DCB=(LRECL=80,RECEM=FB,BLKSIZE=800),
//      VOLUMESER~111111,UNIT=disk
//SYSIN DD *
      BLDG INDEX=A.B.C,ENTRIES1 0,EMPTY,DELETE
```

/\*

```
//STEPB EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD DSNAME=A.B.C(+1),UNIT=disk,DISP=(,CATLG),
// VOLUMESER=222222,SPACE=(TRK,20)
//SYSUT1 DD DATA
```

(input data)

```
//STEP C EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSIN DD DUMMY
//SYSUT2 DD DSNAME=A.B.C(+2),UNIT=disk,DISP=(,CATLG),
//      DCB=(LRECL=80,RECFM=FB,BLKSIZE=1600),
//      VOLUME=SER=222222,SPACE=(TRK,20)
//SYSUT1 DD DATA
```

(input data)

/\*

---

## 8 Sorting Data

There are a number of utility programs available for sorting data. The two most common are ICEMAN and SYNCSORT. Many installations change the name of the program to simply SORT.

---

### 8.1 Sort JCL

Both sort programs use the JCL shown below:

```
//SORT1 EXEC PGM=ICEMAN
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=SYSED.UNSORTED.DATA,
// DISP=SHR
//SORTOUT DD DSN=SYSED.SORTED.DATA,
// DISP=(,CATLG),
// UNIT=SYSDA,
// SPACE=(CYL,5)
//SYSIN DD *
SORT FIELDS=(10,4,CH,A)
```

SYSOUT directs the sort messages to a SYSOUT class. SORTIN is the ddname identifying the input data to be sorted. After sorting, the data is written to the SORTOUT DD statement.

Sometimes the sort program requires you to specify sort work areas. These are unnamed temporary disk data sets with the ddname SORTWKO1, SORTWKO2, AND SORTWKO3. Consult the reference manual for your sort program to determine when you must include these statements.

## 8.2 Sort Control Statements

The SYSIN DD statement points to a control statement which tells the sort program:

- C     What fields to sort.
- C     What format the data is in.
- C     What order to sort into.

9 GDG

```

File Edit Confirm Menu Utilities Compilers Test Help
EDIT      TR021.USER.CNTL(DEFGDG) - 01.00          Columns 00001 00071
Command ==>                                     Scroll ==> CSR
***** ***** Top of Data *****
000010 //TR021A  JOB ('JCL CLASS'),'JCLCLASS',MSGCLASS=X,CLASS=A,NOTIFY=TR021
000016 //STEP1  EXEC PGM=JDCAMS
000017 //SYSPRINT DD SYSOUT=*
000020 //SYSIN   DD *
000030          DEF GDG-
000040          (NAME(TR021.CLASS.MASTER) -
000050          LIMIT(3)
000060          SCRATCH)
000070 /*
000080 //
***** ***** Bottom of Data *****

```

```

Menu Options View Utilities Compilers Help
DSLIST - Data Sets Matching TR021                                     Row 1 of 12

```

```

File Edit Confirm Menu Utilities Compilers Test Help
EDIT      TR021.USER.CNTL(COPY1STG) - 01.00       Columns 00001 00072
Command ==>                                     Scroll ==> CSR
***** ***** Top of Data *****
000010 //TR021A  JOB ('JCL CLASS'),'JCLCLASS',MSGCLASS=X,CLASS=A,NOTIFY=TR021
000012 /**
000013 /**      COPIES AN INSTREAM FILE TO 1ST  GDG  FILE
000014 /**
000015 /**
000016 //STEP1  EXEC PGM=IEBGENER
000017 //SYSPRINT DD SYSOUT=*
000019 //SYSUT2  DD DSN=TR021.CLASS.MASTER(+1),DISP=(NEW,CATLG),
000020 // SPACE=(TRK,1)
000030 //SYSIN   DD DUMMY
000050 //SYSUT1  DD *
000060          Z1001
000070          X3000
000080          X3000
000090          Y7777

```

```

Menu Options View Utilities Compilers Help
DSLIST - Data Sets Matching TR021                                     Row 1 of 12

```

```

File Edit Confirm Menu Utilities Compilers Test Help
EDIT      TR021.USER.CNTL(COPYG2G) - 01.00          Columns 00001 00072
Command --->                                     Scroll ---> CSR
***** Top of Data *****
000010 //TR021A  JOB ('JCL CLASS'),'JCLCLASS',MSCCLASS=X,CLASS=A,NOTIFY=TR021
000011 //*
000012 //*      COPIES AN INSTREAM FILE TO 1ST  GDG  FILE
000013 //*
000014 //*
000015 //STEP1  EXEC PGM=IEBCENER
000016 //SYSPRINT DD  SYSOUT=*
000017 //SYSUT2  DD  DSN=TR021.CLASS.MASTER(+1),DISP=(NEW,CATLG),
000018 // SPACE=(TRK,1)
000019 //SYSIN   DD  DUMMY
000020 //SYSUT1  DD  DSN=TR021.CLASS.MASTER(+0),DISP=(OLD,CATLG)
***** Bottom of Data *****

```

```

Menu Options View Utilities Compilers Help
DSLIST  Data Sets Matching TR021                      Row 1 of 12

```

```

File Edit Confirm Menu Utilities Compilers Test Help
EDIT      TR021.USER.CNTL(COPYG2G)  01.00          Columns 00001 00072
Menu Options View Utilities Compilers Help

```

```

DSLIST - Data Sets Matching TR021                      Row 1 of 12
Command ===>                                     Scroll ===> CSR
Command - Enter "/" to select action                Message                Volume
-----
TR021                                               *ALIAS
TR021.CLASS.MASTER                                ??????
TR021.CLASS.MASTER.G0001V00                       PTS004
TR021.CLASS.MASTER.G0002V00                       PTS003
TR021.FACCLASS.EMPLOYEE                           PTS004
TR021.FACCLASS.FORMATS                             PTS006
TR021.FACCLASS.XREF                                PTS008
TR021.ISR0001.BACKUP                               PTS010
TR021.JCLLIB.CNTL                                  PTS003
TR021.SPFTEMP0.CNTL                                PTS008
TR021.SYSP.TSPF.ISPPROP                            PTS005
TR021.USER.CNTL                                    PTS002
***** End of Data Set list *****

```