

MAINFRAME

FILES - DATASETS - FLAT - PS(PHYSICAL SEQUENTIAL)

FOLDER - PARTITIONED DATASET - PDS,PDSE,LIBRARY(LOAD MODULES)

FILES - MEMBERS

JOB - PROGRAM

MAINFRAME

SOFTWARES - SUBSYSTEMS

- 1. TSO -time sharing options· Logon*
- 2. ISPF/PDF -INTERACTIVE SYSTEM PRODUCTIVITY FACILITY/PROGRAM DEVELOPMENT FACILITY·*
- 3. JES - JOB ENTRY SUBSYSTEM·*
- 4. DB2 -RDBMS MAINFRAME*
- 5. CICS - OLTP*

BATCH PROCESSING - MAINFRAME

ONLINE TRANSACTION PROCESSING -

MEMORY·

MAIN - RAM

SECONDARY

- 1. DASD - DIRECT ACCESS STORAGE DEVICE - 3390 HDD*
- 2. TAPE - AUDIO/VHS BACKUP·*

PARTITIONED INTO VOLUMES/SERIAL NUMBERS

Data set allocation parameters.

1. DSN

HLQ.Q2.Q3.Q4.Q5.Q6.Q7

HIGH LEVEL Q → USER MF ID → ARI001

Qualifiers → 8 chars max. first char must be an alphabet.

Alphanumeric

Every q must be separated by a period

Max 7 qs.

Total number of chars : 44.

OZAGSI.ALWYN.OUTPUT.DATA7.P5

2. SPACE UNITS.

BYTES, MB,KB, TRACKS, CYL

MB/TRACK. TRACKS/CYL

3. PRIMARY QUANTITY.

It is the initial amount of space units.

3

4. Secondary quantity.

It represents the number of space units that can be allocated **WHEN THE PREVIOUSLY ALLOCATED SPACE UNITS ARE FULLY UTILIZED.**

2 x 15 TIMES

SPACE ABEND

5. Directory blocks: folder(PDS,LIBRARY,PDSE)

It represents the max number of members inside the folder.

$(N \times 6) - 1$

6. Volume/serial (optional)

7. Logical REcord Length(LRECL)

The number of chars for each.

Min = 1, max num=32kb

Job/cobol program - 80

Dataset - depending on the num of chars per record

8. BLKSIZE = multiples of LRECL

Unit/min amount of data transfer between secondary and the main memory.

9. RECFM -record format

F - FIXED, FB- FIXED BLOCK, FBA - ADDRESSING

V - VARYING, VB - VARYING BLOCK , VBA- ADDRESSING

U - UNFORMATTED/UNKNOWN → LOAD MODULES.

10. DSORG/DS TYPE →

EMPTY → PS

PDS, PDSE, LIBRARY

11. EXPIRY

DAYS 200.

ISPF

3 → UTILITIES

2 → ALLOCATE

A parameters.

File commands:

D - DELETES

R - RENAMES

E - OPEN IN EDIT MODE

V- OPEN IN VIEW MODE (CHANGES CAN BE DONE, BUT CANT BE SAVED)

B - OPEN IN BROWSE MODE (READ ONLY)

I - INFORMATION

S - SHORT INFO

PDS

C - COPY A MEMBER

Z - COMPRESS

TSO EDIT COMMANDS

EDIT RECORDS

I - INSERT

IX - X NUMBER OF LINES WILL BE INSERTED

D - DELETE

DX - X NUMBER OF LINES WILL BE DELETED.

R - REPEATS A LINE

RX - REPEATED X NUMBER OF TIMES.

C - COPY A RECORD/LINE

A-AFTER

B-BEFORE

CX - COPY X NUMBER OF LINES

M- MOVE (CUT A LINE)

A - AFTER

B-BEFORE

MX - CUT X NUMBER OF LINES

X - HIDES A RECORD

XX - HIDES X NUMBER OF LINES.

) → moves the line 2 positions to the right

)x → moves the line with x number of positions

(→ move the record to the left.

LC → LOWER CASE

LCX →

UC → UPPER CASE

UCX →

TS - SPLITS a record across next line

TF - text flow →

MASK → It allows the user to give titles for a the fields

COLS → DISPLAY THE COLUMN NUMBERS

BNDS → BOUNDARY INSIDE WHICH THE COMMANDS WILL WORK.

< >

LABELS → IDENTIFYING RECORDS BY LABELING IT. “.CHAR”

Commands.

- *SAVE*
- *END/EXIT → PREVIOUS SCREEN*
- *RESET(RES) → REFRESHING THE SCREEN.*

- FIND → F "STRN" , F ALL 'STRNG", F ALL ' STRNG' STPOS ENDPOS
- CHANGE → C 'NEW YORK' 'INDIA'
- SORT
- SUBMIT
- START → STARTS ANOTHER PARALLEL SCREEN. F9 TO SWAP
- PROFILE
 - CAPS ON
 - HILIGHT
 - UNNUM

Allocation of a PDS

JOBS.

Group of tasks.

Column definition.

Lrecl=80 only, recfm=fb only

1,2 → identifier field → //

3-10 → Name field (max 8 chars. Starting with alphabet, then alpha numeric)

11 empty

12-15 operation field → JOB,EXEC,DD

16—PARAMETER FIELD

72-80 MUST NOT BE USED.

3 statements.

1. Job card (1)

PARAMETERS.

POSITIONAL(OPTIONAL)

AUTHOR NAME "ALWYN"

ACCOUNTING INFO (BNYM, CREDIT)

KEYWORD

NOTIFY(MANDATORY) → =&SYSUID

CLASS=A-Z, 0-9 → CLASSIFICATION

MSGCLASS= A-Z,0-9 → FORMAT OF THE REPORT

MSGLEVEL=(X,Y)(1,1)

X = what type (0,1,2)

Y = when do you want the report to be generated (0,1)

PRTY= PRIORITY → 1-15

TIME=(M,S) Requesting to allot time period for this job.

Time=2 → min

Time=(,3)→ 3 sec

REGION=OM (max allowable memory space in the execution region)

COND = LATER

TYPRUN=SCAN/HOLD

SCAN→ Scans the job for syntax errors. It will not execute.

HOLD → Stays in the job queue until authorized user releases it for execution.

RESTART=STEPNAME

In a multiple step job, if you want execute from a particular step, skipping the previous steps, you can mention RESTART=STEPNAME.

2. Job step (255)

//STEPNAME EXEC PGM=UTILITY,PARAMETERS
UTILITY

SYSTEM

USER

IEFBR14 → DUMMY, BR14

IEBGENER → COPY FROM A PS TO A PS

IEBCOPY → COPY FROM PDS TO A PDS

IEBEDIT → EDIT

SORT → DATA PROCESSING.

IDCAMS → CATALOG DATASETS

PARAMETERS.(OPTIONALS)

REGION= means the same as in jobcard but this is for individual step.

TIME= means the same as in jobcard but this is for individual step.

COND= later

PARM= COMPILER OPTIONS,INPUT TO COBOL.

3. Dd statements(3273)

a. USER DEFINED

//DDNAME DD

DDNAME: DEPENDS ON THE UTILITY

IEFBR14,IDCAMS → USER DEFINED

DD1, DD2, DD3

SORT → SORTIN, SORTIN01,SORTIN02 → INPUT

SORTOUT, SORTOF01,SORTOF02... → OUTPUT

IEBGENER/IEBCOPY → SYSUT1 → INPUT

SYSUT2→ OUTPUT

//DDNAME DD PARAMETERS.

MANDATORY:

DSN=DATASET NAME.

DISP=(D1,D2,D3)

D1 -> CURRENT STATUS OF THE DATASET

NEW → It allocates/creates a new dataset

OLD → the DS exist, if you want read from it.

If you want write into it.

SHR → the DS exist, if you want read from it.

If you want write into it,

It writes data into the dataset by emptying it first.

MOD → if the exists, if it is used for output file,

It appends the new data

If the file doesn't exist, it acts as *NEW*

D2 → status of the DS, after the successful execution of the job.

KEEP → LOCAL DIRECTORY

DELETE → PHYSICALLY REMOVES THE FILE FROM THE MEMORY.

CATLG → THE DS IS MOVED TO MAIN DIRECTORY

UNCATLG → REMOVES FROM THE CATLG AND PUT IN LOCAL.

PASS → WHEN A DATASET USED IN A STEP MIGHT ALSO BE USED IN CONSECUTIVE STEPS, *PASS* WILL RETAIN THE DATSET IN THE EXECUTION REGION.

D3 → status of the DS, after A FAILURE of the job.

KEEP

DELETE

CATLG

UNCATLG

DEFAULT VALUES →

DISP=NEW → *DELETE,DELETE*

DISP=OLD,SHR → *KEEP,KEEP*

DISP=MOD → *DEPENDS.*

When *DISP* allocates a new dataset.

SPACE=(SU,(PQ,SQ,DB),CONTIG,RLSE,ROUND)

SU → *TRK,CYL,*

PQ=2

SQ=3

DB=2 → *ONLY PDS.*

CONTIG → *ALLOCATE CONTINUOUS SPACE UNITS*

RLSE → *Release the unused space automatically.*

ROUND → *Release the memory rounding it to SPACE UNITS.*

DCB(directory Control Block)=(LRECL=80,BLKSIZE=800,RECFM=FB,DSORG=PS)

DSORG= PS → *PS*

PO → *PDS*

LIB → *LIBRARY*

b. *SYSTEM(MANDATORY)*

OUTPUT

//SYSPRINT DD SYSOUT= →*

SYSPRINT → REPORTS

SYSOUT= → PUT THAT IN THE SPOOL*

*//SYSOUT DD SYSOUT=**

SYSOUT → SORT CARD ERRORS

INPUT

*//SYSIN DD **

SORT FIELDS=(1,4,CH,A)

*/**

SYSIN DD DUMMY → no inputs/control cards are given in the job.

*SYSIN DD *, star represents that inputs(data,control cards) are in the next line till the delimiter.*

Note: When a data or the control cards are given in the job, it is called as INSTREAM DATA/CARDS.

When a job is submitted

A JOBID is generated by the JES.

The status of the job will be represented in 2 ways.

ABEND CODES - S806, SOC4, SOC7, 722,

CONDITION CODE/RETURN CODE

CC/RC → numerical representation of the status of the job.

0000 → successful

0004 → successful but with warning/info

>=0008, 12,16,20,22 ...4095 → severe errors.

JCL ERROR/INTERNAL ERRORS

SPOOL → a small buffer where the reports of all the jobs are deposited. Simultaneous Peripheral Output OnLine.

IEBGENER

INPUT DD NAME:SYSUT1. INSTREAM DATA/INPUT DATASET
OUTPUT DD NAME : SYSUT2. SPOOL/DATASET.

Note: Whenever a new dataset is about to be created in a Job, ensure that that dataset doesn't exist.

Predeletion → 1. If the file exist, delete.

If the file doesn't exist, create it and delete.

BACKWARD REFERENCE.

It is a technique in which the dcb parameter of a ddname can be made to refer a dcb parameter mentioned in any previous ddnames.

If it refers to a ddname within the step

DCB=.DDNAME*

If it refers to a ddname of previous step

DCB=.STEPNAME.DDNAME.*

Handson 1.

Isfp 3.2

Create a ps → hlq.data.ps

1234 QWERTY CSE 79.23 87.63

STORE 6 RECORDS IN THE BOVE FORMAT

WRITE A JOB.

STEP000 PREDELETE

HLQ.DATA.OUTPUT.PS1.

HLQ.DATA.OUTPUT.FINAL

STEP001

COPY THE DATA FROM HLQ.DATA.PS INTO HLQ.DATA.OUTPUT.PS1.

INPUT : ALREADY EXISTS

OUTPUT: ALLOCATE IN THE STEP.

STEP002

COPY DATA FROM HLQ.DATA.OUTPUT.PS1 INTO HLQ.DATA.OUTPUT.FINAL

Note: If the data is successfully copied into hlq.data.final ps, delete the hlq.data.output.ps1.

If not, catlg the hlq.data.output.ps1 dataset.

Set parameter

Empty file handling.

IF STRUCTURE

 // IF (STENAME·RC OPRTR VALUE) THEN

 //STEP1

 // ELSE

 //STEP2

 // ENDIF

21-09-2023

Agenda:

IEBCOPY

CONTROL LIBRARY

PROCEDURES

SORT UTILITY

CONTROL LIBRARY:

It is a technique in which the control statements(cards/sort cards) are written in a different pds member and that PDS member can be mentioned in the SYSIN DD DSN= of the JOB.

The PDS → control Library.

The member → control book.

IEBCOPY.

INPUT DD NAME → SYSUT1

OUTPUT DDNAME → SYSUT2

COPIES ALL THE MEMBERS FROM THE INPUT PDS INTO OUTPUT PDS.

If certain members to be selected or excluded, control card is used.

The ddnames are of the user's choice.

Sysin dd *

COPY OUTDD=DD2

INDD=DD1

SELECT/EXCLUDE MEMBER=(list of members separated by comma)

Note: If the input PDS and the output PDS are the same PDS, then this job would COMPRESS the PDS.

PROCEDURES.

It is a collection reusable steps that can be invoked by a job step.

EXEC PROC=PROCNAME

A Procedure can have a max of 15 steps.

Syntax:

PROC PROC1

PSTEP1

PSTEP2

PEND

INSTREAM

Defined in the job where it is invoked. i.e The job and the procedure are written in the same member. Scope: only within the job.

The definition of the procedure must be done AFTER THE JOB CARD AND BEFORE THE FIRST JOB STEP.

CATALOG

Defined in another pds member. Any job from any pds can invoke this procedure.

Note: The name of the member in which the procedure is defined and the name of the procedure MUST BE SAME.

The location of the catalog procedure is mentioned with JCLLIB ORDER=PDS

After the job card and before the first Job step.

Overriding dd statements.

When the user wants to dummy a dd name mentioned in the procedure and wants to give a different dataset from the invoking job, it can be done by the technique called overriding dd statement.

Syntax: stepname.ddname

SORT

1. *SORT*
2. *REFORMAT(CHANING THE FIELDS,INTRODUCING NEW FIELDS,REMOVING FIELDS, ARITHMATIC OPERATIONS ETC)*
3. *SUM UP VALUES.*
4. *ELIMINATE DUPLICATE RECORDS*
5. *SPLIT RECORDS INTO MUTIPLE OUTPUT FILES.*
6. *MERGE DATA*
7. *CHOOSE RECORDS BY CONDITIONS(FILTERING)*
8. *GENERATE REPORTS.*

DEFAULT DD NAMES.

INPUT → SORTIN, SORTIN01, SORTIN02 ETC

OUTPUT → SORTOUT, SORTOF01, SORTOF02...

SORTWK01 → TEMPORARY.(OPTIONAL)

*SYSOUT DD SYSOUT=**

*SYSIN DD **

SORT CARDS

*/**

1001 TOMMY CHENNAI 80000

1234 JERRY DISNEY 70000

TO SORT RECORDS IN SOME ORDER BASED ON A FIELD

SORT FIELDS=(STPOS,LENGTH,TYPE,ORDER)

STPOS → COL NUMBER OF THE FIELD BASED ON WHICH IT IS TO BE SORTED.

LENGTH → NUMBER OF CHARS THAT MAKES THE FIELD.

TYPE → CH→CHAR, ZD → ZONED DECIMAL, PD→ PACKED DECIMAL

ORDER → A→ ASCENDING, D→ DESCENDING

SORT BASED ON MORE THAN ONE COLUMN

SORT FIELDS=(STPOS,LENGTH,TYPE,ORDER, STPOS,LENGTH,TYPE,ORDER)

ELIMINATING DUPLICATE RECORDS.

SORT FIELDS=(1,4,CH,A)

SUM FIELDS=NONE

Note: The sum fields will be applied on data based duplicate values in the field mentioned in the SORT FIELDS.

Summing up the values in (22,6) if duplicates in(12,9) another field

```
//SYSIN DD *  
    SORT FIELDS=(12,9,CH,A)  
    SUM FIELDS=(22,6,ZD)  
/*
```

Choosing records based on conditions

LOGICAL → AND, OR

COMPARATIVE → LT,GT,GE,LE,EQ,NE

```
//SYSIN DD *
```

```
    SORT FIELDS=(1,4,CH,A)
```

```
    SUM FIELDS=NONE
```

```
    INCLUDE/OMIT COND=(STPOS,LENGTH,CH,EQ,C'STRNG')  
                        = (STPOS,LENGTH,ZD,EQ,12345)
```

```
/*
```

FOR A RANGE OF SALARIES

```
SORT FIELDS=(1,4,CH,A)
```

```
/* CHOOSING RECORDS WHITH SALARY GE 050000 AND LE 080000
```

```
INCLUDE COND=(22,6,ZD,GE,050000,AND, 22,6,ZD,LE,080000)
```

```
/* CHOOSING RECORDS WITH LOCATION AS NEW JERSY OR NEW YORK
```

```
INCLUDE COND=(12,9,CH,EQ,C'NEW JERSY',OR,12,8,CH,EQ,C'NEW YORK')
```

```
/* CHOOSING RECORDS WITH LOCATION AS NEW YORK AND SALARY GE 05000
```

```
INCLUDE COND=(12,8,CH,EQ,C'NEW YORK', AND,22,6,ZD,GE 050000)
```

22-09-2023.

AGENDA

SORT

GDG

ASSGINMENT.

SORT.

REFORMATTING.

REORGANIZING THE FIELDS IN THE OUTPUT

PICKING AND CHOOSING THE FIELDS TO THE OUTPUT

ARITMATIC OPERATIONS

INTRODUCING NEW FIELDS IN THE OUTPUT.

INREC WILL ACT BEFORE SORTING.

INREC FIELDS → LIST ALL THE FIELDS THAT YOU NEED.(the fields that are not mentioned, will not be brought to the output)

INREC OVERLAY → BE DEFAULT ALL THE FIELDS WILL BE BROUGHT TO THE OUTPUT.

OUTREC WILL ACT AFTER SORTING

OUTREC FIELDS → LIST ALL THE FIELDS THAT YOU NEED.(the fields that are not mentioned, will not be brought to the output)

OUTREC OVERLAY→ BE DEFAULT, ALL THE FIELDS WILL BE BROUGHT TO THE OUTPUT.

OUTREC FIELDS=(<POS IN THE OUTPUT>:STPOS OF THE FIELD YOU WANT,LENGTH)

Exercise:

Input sample data:

1001 tommy Chennai 80000

Write a SORT job to generate email id for each record. The email id consists of first 3 chars of the name and the first 3 chars of location and this string '@revature.com'. The email must be printed at the 30th column.

The expected output

1001 tommy Chennai 80000 tomche@revataure.com

Outrec fields=()

ADD, MUL,SUB,DIV

SPLITTING RECORDS FROM A SINGLE FILE TO MULTIPLE OUTPUT FILES.

SORTIN DD

SORTOUT DD

WHEN MORE THAN 1 OUTPUT FILES → SORTOF01, SORTOF02 ETC

SPLIT BASED WHAT?

1. VALUE IN A COLUMN.
//JOB CARD
//STEP1 EXEC PGM= SORT
//SORINT DD
//SORTOF01 DD DSN
//SORTOF02 DD DSN

```
//SYSIN DD *  
SORT FIELDS=COPY  
OUTFIL FNAMES=01,INCLUDE=()  
OUTFIL FNAMES=02,INCLUDE=()  
2. SEQUENCE OF THE RECORDS  
SORT FIELDS=COPY  
OUTFIL FNAMES=01,STARTREC=1,ENDREC=2  
OUTFIL FNAMES=02,STARTREC=3,ENDREC=4
```

Topics pending:

COND parameters

Sort if then when

SORT MERGE

GDG

SORT ARITHMETIC OPERATION WITH UFF AND SFF