

**Chapter
2**

**INTERNAL
SORTS**

*Get on the
Fast Track!*



TM

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

Objectives

You will learn:

- Sorting - role and purpose.
- Advantages and tradeoffs associated with an internal and external sort.
- How to code the Data Division for a sort.
- Coding the SORT options.
- RELEASE and RETURN statements.
- SORT verb.
- Coding input and output procedures.
- SORT - Output PROC.
- DFSORT - Data Facility Sort.
- Controlling sort behavior.

1 Sorting - What is it?

Sorting is a technique in which data can be processed in a sequence different from which it was initially stored.

1.1 Sorting Data Files

Sorting of sequential data files are typically accomplished in one of two ways:

- A stand-alone sort utility which requires the programmer to know the parameters of the utility program.
- The COBOL verb SORT which requires the programmer to know only COBOL.

Records can be arranged in a particular sequence by using a SORT statement. The SORT accepts input from a file or an internal procedure and produces output to a file or an internal procedure in a requested sequence.

Records can be added, deleted, or changed before or after they are sorted.

A program can contain any number of sort operations. They can be the same operation performed many times or different operations. However, one operation must finish before another begins.

The general steps for performing a sort or merge are:

- Describe the sort file to be used for sorting or merging.
- Describe the input to be sorted.
 - If the records are to be processed prior to being sorted, code an input procedure.
- Describe the output from the sort.
 - If the records are to be processed after the sort has been performed, then code an output procedure.
- Request the sort or merge.
- Determine whether the sort operation was successful.

During the sorting of a file, all of the records in the file are ordered according to the contents of one or more keys in each record. The records can be sorted in either ascending or descending order on each key.

If there are multiple keys, the records are first sorted according to the content of primary key, then according to the content of the second key, and so on.

2 Environment Division Considerations

A SELECT statement must be specified for the sort file. It is used to assign a name to the sort file and designate a DD name for any work datasets which are needed.

2.1 Sort SELECT Format

```
SELECT sort-file-name  
    ASSIGN TO [integer] assignment-name-1
```

3 Data Division Considerations

In order for a sort to be performed in a COBOL program, the program must include FD: file description entries for files that are input to or output from the sort.

In addition, the sort file itself must be defined in a SD: sort description entry.

3.1 Sort File Description

```
SD    sort-file-name
      [RECORD CONTAINS [integer-1 TO] integer-2 CHARACTERS]
      DATA RECORD IS sort-record-name.

01    sort-record-name
```

RECORD CONTAINS n	Size of logical records	fixed length
	The integer size indicates the number of bytes in a record regardless as to the USAGE of the data items in the record.	
RECORD CONTAINS n TO m	Size of logical records	variable length
	The integer sizes indicate the number of bytes in a record regardless as to the USAGE of the data items in the record.	
DATA RECORDS	Names of records associated with the file.	Comments only.

4 Procedure Division Considerations

The Procedure Division must contain a SORT statement to describe the sort operation. Input and output sort procedure names are optional.

4.1 SORT Statement Format

The sort-file-name identifies a temporary work file that the SORT process uses for the sort. It is defined in the FILE SECTION using an SD: Sort Description entry. Even though the work file is a temporary file, it must still have an associated SELECT and ASSIGN clause in the ENVIRONMENT DIVISION.

The sort-file-name file is a sequential file with an organization of RECORD SEQUENTIAL. Since this is the default organization is it usually is omitted.

Each data-name-n identifies a field in the record of the work file. The sorted file will be in sequence on this key field(s). When more than one data-name-n is specified, the keys decrease in significance from left to right. The leftmost key is most significant; rightmost key is least significant.

The file-name-2 and file-name-3, are the names of the input and output files respectively.

Section is no longer required in the Procedure Division.

```

                |DESCENDING|
SORT sort-file-name ON <          > KEY <data-name-1>
                | ASCENDING|
                                <data-name-2> . . .

ON      |DESCENDING|
        <          > KEY <data-name-3> . . .] . . .
        | ASCENDING|

|INPUT PROCEDURE IS section-name-1 [THRU section-name-2]|
|<          >|
|USING file-name-2          |

|OUTPUT PROCEDURE IS section-name-3 [THRU section-name-4]|
|<          >|
|GIVING file-name-3          |
    
```

5 RELEASE Statement Format

The RELEASE statement transfers records from an input/output area to the initial phase of a sorting operation. Within an INPUT PROCEDURE, at least one RELEASE statement must be specified.

When the RELEASE statement is executed, the current content of record-name-1 is placed in the sort file. This makes the record available to the initial phase of the sorting operation.

```
RELEASE sort-record-name [FROM identifier]
```

5.1 RETURN Statement Format

The RETURN statement transfers records from the final phase of a sorting or merging operation to an OUTPUT PROCEDURE.

The RETURN statement can be used only within the range of an OUTPUT PROCEDURE associated with a SORT.

Within an OUTPUT PROCEDURE, at least one RETURN statement must be specified. When the RETURN statement is executed, the next record from file-name-1 is made available for processing by the OUTPUT PROCEDURE.

```
RETURN sort-file-name [INTO identifier]
```

```
AT END imperative-statement
```

6 Programming Example

6.1 COBOL SORT Verb

These examples demonstrate the use of the COBOL SORT verb.

Example 1: SORT verb with the USING and GIVING options.

```
ID DIVISION.
.
ENVIRONMENT DIVISION.
.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT TRANS-FILE-IN  ASSIGN TO INTRANS.
    SELECT TRANS-FILE-OUT ASSIGN TO INTRANS.
    SELECT SORT-WORK      ASSIGN TO WORKFILE.

DATA DIVISION.
FILE SECTION.

FD  TRANS-FILE-IN
.
.
01  TRANS-IN-RECORD          PIC X(80).

FD  TRANS-FILE-OUT
.
.
01  TRANS-OUT-RECORD        PIC X(80).

SD  SORT-WORK
    DATA RECORD IS SORT-RECORD.

01  SORT-RECORD.
    05  SD-SS-NUMBER          PIC X(9).
    05  SD-DEPT-NO           PIC XX.
    05  SD-HOURS             PIC 99.
    05  FILLER                PIC X(63).
    05  SD-TRANS-CODE        PIC XX.

PROCEDURE DIVISION.

    SORT SORT-WORK ON ASCENDING SD-DEPT-NO
        ON ASCENDING KEY SD-SS-NUMBER
        USING TRANS-FILE-IN
        GIVING TRANS-FILE-OUT.

    STOP RUN.
```

Example 2: Sort VERB with an input and output procedure.

```
ID DIVISION.
PROGRAM-ID. SORTLAB1.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT GOVINP ASSIGN TO GOVINP.
    SELECT GOVSORT ASSIGN TO GOVSORT.

DATA DIVISION.
FILE SECTION.
FD GOVINP.
01 GOVINP-RCD                PIC X(100).

SD GOVSORT.
01 GOVSORT-RCD.
    05 FILLER                PIC X(57).
    05 GR-YEAR-ENDING        PIC X(4).
    05 FILLER                PIC X(39).

WORKING-STORAGE SECTION.
01 GOV-RECORD.
    05 GR-NAME                PIC X(24).
    05 GR-STATE                PIC X(15).
    05 GR-PARTY                PIC X(11).
    05 GR-START-YEAR          PIC X(4).
    05 FILLER                 PIC X.
    05 GR-END-YEAR            PIC X(4).
    05 FILLER                 PIC X(42).

01 GOV-EOF-SW                PIC X VALUE 'N'.
01 GOVSORT-EOF-SW            PIC X VALUE 'N'.
01 WS-PRIOR-END-YEAR         PIC X(4) VALUE LOW-VALUES.

PROCEDURE DIVISION.
000-MAINLINE.
    SORT GOVSORT ON ASCENDING KEY GR-YEAR-ENDING
        INPUT PROCEDURE 100-READ-DATA
        OUTPUT PROCEDURE 200-PROCESS-DATA
    STOP RUN.

100-READ-DATA.
    OPEN INPUT GOVINP

    READ GOVINP INTO GOV-RECORD
    AT END
        MOVE 'Y' TO GOV-EOF-SW
    END-READ
```

```
PERFORM UNTIL GOV-EOF-SW = 'Y'
  RELEASE GOVSORT-RCD FROM GOV-RECORD
  READ GOVINP INTO GOV-RECORD
  AT END
    MOVE 'Y' TO GOV-EOF-SW
  END-READ

END-PERFORM
CLOSE GOVINP
.
```

200-PROCESS-DATA.

```
RETURN GOVSORT INTO GOV-RECORD
  AT END
    MOVE 'Y' TO GOVSORT-EOF-SW
  END-RETURN
MOVE GR-END-YEAR TO WS-PRIOR-END-YEAR

PERFORM UNTIL GOVSORT-EOF-SW = 'Y'
  PERFORM 300-REPORT
  RETURN GOVSORT INTO GOV-RECORD
  AT END
    MOVE 'Y' TO GOVSORT-EOF-SW
  END-RETURN

END-PERFORM
.
```

300-REPORT.

```
IF GR-END-YEAR = WS-PRIOR-END-YEAR
  CONTINUE
ELSE
  DISPLAY SPACES
END-IF
MOVE GR-END-YEAR TO WS-PRIOR-END-YEAR
DISPLAY GOVSORT-RCD
.
```

6.2 SORT - Output PROC

```
ID DIVISION.
PROGRAM-ID.  SORTLAB2.
ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT BENINP  ASSIGN TO BENINP.
    SELECT BENSORT ASSIGN TO BENSORT.

DATA DIVISION.

FILE SECTION.
FD  BENINP.
01  BENINP-RCD                PIC X(110).

SD  BENSORT.
01  BENSORT-RCD.
    05 FILLER                  PIC X(19).
    05 CL-DATE.
        10 BR-MM              PIC 99.
        10 BR-DD              PIC 99.
        10 BR-YYYY            PIC 9(04).
    05 FILLER                  PIC X(73).

WORKING-STORAGE SECTION.
01  BENEFITS-REC.
    05 SSN.
        10 SSN1                PIC 999.
        10 SSN2                PIC 99.
        10 SSN3                PIC 9999.
    05 BRANCH                  PIC X(05).
    05 CLAIM-NUM               PIC X(05).
    05 CLAIM-DATE.
        10 CL-MM               PIC 99.
        10 CL-DD               PIC 99.
        10 CL-YYYY             PIC 9(04).
    05 CLAIM-TERM-DATE.
        10 CL-T-MM             PIC 99.
        10 CL-T-DD             PIC 99.
        10 CL-T-YYYY           PIC 9(04).
    05 WORKER-NAME             PIC X(20).
    05 BENE-NAME               PIC X(20).
    05 BIC                     PIC X(02).
    05 WORKER-SEX              PIC X(01).
    05 YRS-EMPLOY              PIC 9(02).
    05 DOB.
        15 DOB-MM              PIC 99.
        15 DOB-DD              PIC 99.
        15 DOB-YY              PIC 9(4).
    05 CHILD                   PIC 9(02).
    05 PIA                     PIC 9(03)V99.
    05                         PIC X(15).
```

```
01 BEN-EOF-SW          PIC X VALUE 'N'.
01 BENSORT-EOF-SW     PIC X VALUE 'N'.
01 WS-PRIOR-MM        PIC 99.
01 WS-PRIOR-YYYY     PIC 9(4).
01 WS-TOT-MM         PIC S9(5)V99 COMP-3 VALUE ZERO.
01 WS-TOT-YYYY       PIC S9(5)V99 COMP-3 VALUE ZERO.
01 WS-TOT-GRAND      PIC S9(5)V99 COMP-3 VALUE ZERO.
01 WS-PRETTY-NUMBER  PIC ZZ,ZZZ.99 VALUE ZERO.
01 WS-PRETTY-DATE.
   05 WS-PRETTY-MM    PIC 99.
   05 FILLER          PIC X VALUE '/'.
   05 WS-PRETTY-DD    PIC 99.
   05 FILLER          PIC X VALUE '/'.
   05 WS-PRETTY-YYYY PIC 9999.
```

PROCEDURE DIVISION.

000-MAINLINE.

```
    SORT BENSORT ON ASCENDING KEY BR-YYYY
                                   BR-MM
                                   BR-DD
        INPUT PROCEDURE 100-READ-DATA
        OUTPUT PROCEDURE 200-PROCESS-DATA
    IF SORT-RETURN = 0
        CONTINUE
    ELSE
        DISPLAY 'ERROR IN SORT - SORRY'
        MOVE 16 TO RETURN-CODE
        STOP RUN
    END-IF
```

```
MOVE WS-TOT-MM TO WS-PRETTY-NUMBER
DISPLAY SPACES
DISPLAY 'TOTAL FOR MONTH ' WS-PRIOR-MM
      ' WS-PRETTY-NUMBER
MOVE WS-TOT-YYYY TO WS-PRETTY-NUMBER
DISPLAY SPACES
DISPLAY 'TOTAL FOR YEAR ' WS-PRIOR-YYYY
      ' WS-PRETTY-NUMBER
MOVE WS-TOT-GRAND TO WS-PRETTY-NUMBER
DISPLAY SPACES
DISPLAY 'GRAND TOTAL      '
      ' WS-PRETTY-NUMBER
STOP RUN.
```

100-READ-DATA.

```
OPEN INPUT BENINP

READ BENINP INTO BENEFITS-REC
AT END
    MOVE 'Y' TO BEN-EOF-SW
END-READ
```

```
PERFORM UNTIL BEN-EOF-SW = 'Y'
  RELEASE BENSORT-RCD FROM BENEFITS-REC
  READ BENINP INTO BENEFITS-REC
  AT END
    MOVE 'Y' TO BEN-EOF-SW
  END-READ

END-PERFORM
CLOSE BENINP
.
```

200-PROCESS-DATA.

```
RETURN BENSORT INTO BENEFITS-REC
  AT END
    MOVE 'Y' TO BENSORT-EOF-SW
  END-RETURN

MOVE CL-MM TO WS-PRIOR-MM
MOVE CL-YYYY TO WS-PRIOR-YYYY
DISPLAY '                                CLAIM REPORT BY DATE '
DISPLAY SPACES
DISPLAY 'CLAIM DATE      SSN          WORKER NAME
        'PIA'
DISPLAY SPACES

PERFORM UNTIL BENSORT-EOF-SW = 'Y'
  PERFORM 300-REPORT

  ADD PIA TO WS-TOT-MM
  ADD PIA TO WS-TOT-YYYY
  ADD PIA TO WS-TOT-GRAND
  MOVE CL-MM TO WS-PRIOR-MM
  MOVE CL-YYYY TO WS-PRIOR-YYYY

  RETURN BENSORT INTO BENEFITS-REC
  AT END
    MOVE 'Y' TO BENSORT-EOF-SW
  END-RETURN

END-PERFORM
.
```

300-REPORT.

```
IF CL-MM = WS-PRIOR-MM
  CONTINUE
ELSE

  MOVE WS-TOT-MM TO WS-PRETTY-NUMBER
  DISPLAY 'TOTAL FOR MONTH ' WS-PRIOR-MM
        '                                ' WS-PRETTY-NUMBER

  DISPLAY SPACES
  MOVE 0 TO WS-TOT-MM
END-IF

IF CL-YYYY = WS-PRIOR-YYYY
  CONTINUE
ELSE
```

```
MOVE WS-TOT-YYYY TO WS-PRETTY-NUMBER
DISPLAY 'TOTAL FOR YEAR ' WS-PRIOR-YYYY
      '                               ' WS-PRETTY-NUMBER
DISPLAY SPACES
MOVE 0 TO WS-TOT-YYYY
END-IF

MOVE PIA TO WS-PRETTY-NUMBER
MOVE CL-MM TO WS-PRETTY-MM
MOVE CL-DD TO WS-PRETTY-DD
MOVE CL-YYYY TO WS-PRETTY-YYYY
DISPLAY WS-PRETTY-DATE ' '
      SSN              ' '
      WORKER-NAME      ' '
      WS-PRETTY-NUMBER
```

7 Faster Sort Facility

When the DFSORT - Data Facility Sort, is used it may be possible to realize a faster sort.

This improvement is derived from DFSORT performing the related input/output operations.

The FASTSRT compiler option will need to be specified.

- When there is only one input file, do not specify declaratives: label or exceptional error.
- For fast sorting the input, specify the USING option instead of an input procedure.
- For fast sorting the output, specify the GIVING option instead of an output procedure.

7.1 Example: Fastsort

In a simple SORT example, the standard COBOL statements would be:

```
SELECT SORT-INPUT ASSIGN TO INFILE ....
SELECT SORT-OUTPUT ASSIGN TO OUTFILE ....

SORT SORT-FILE ON ASCENDING KEY sort-key
  USING SORT-INPUT
  GIVING SORT-OUTPUT
```

The standard JCL would be:

```
SELECT SORT-INPUT ASSIGN TO INFILE ....
SELECT SORT-OUTPUT ASSIGN TO OUTFILE ....

SORT SORT-FILE ON ASCENDING KEY sort-key
  USING SORT-INPUT
  GIVING SORT-OUTPUT
```

8 Defining Sort Datasets

In order to use SORT under the z/OS operating system, code DD statements in the runtime JCL:

Sort or merge work	SORTWK01, SORTWK02, SORTWK03, . . ., SORTWKnn where nn is 99 or less.
SYSOUT	Define for sort diagnostic messages, unless the data-set name is changed.
SORTCKPT	Define if the sort is to take checkpoints.
Input and output	Define input and output datasets.
SORTLIB	Define the library that contains the sort modules: SYS1.SORTLIB.

9 Sort - Determining Successful

The SORT utility returns a completion code after each sort has finished.

0	successful completion.
16	unsuccessful completion.

The completion code is stored in the SORT-RETURN special register.

A test should be performed for successful completion after each SORT or MERGE statement.

Example:

```
SORT SORT-WORK-2
  ON ASCENDING KEY SORT-KEY
  INPUT PROCEDURE IS 600-SORT3-INPUT-PROC
  OUTPUT PROCEDURE IS 700-SORT3-OUTPUT-PROC.

IF SORT-RETURN = 0
  CONTINUE
ELSE
  DISPLAY `SORT ENDED ABNORMALLY. SORT-RETURN = ` SORT-RETURN.
```

If SORT-RETURN is not referenced anywhere in a program, the COBOL run time tests the completion code. If it is 16, COBOL issues a runtime diagnostic message.

By default, SORT diagnostic messages are sent to the SYSOUT dataset.

10 Stopping a Sort

In order to stop a sort operation, move the integer 16 into the SORT-RETURN special register.

```
MOVE 16 TO RETURN-CODE
```

Move 16 into the register using a MOVE in an input or output procedure. Sort or merge processing will be stopped immediately after the next RELEASE or RETURN statement is performed.

Control then returns to the statement following the SORT or MERGE statement.

11 Controlling Sort Behavior

To set or test	Use this special register or compiler option
Amount of main storage to be reserved.	SORT-CORE-SIZE special register.
Amount of main storage to be used.	SORT-CORE-SIZE special register.
Modal length of records in a file with variable-length records.	SORT-MODE-SIZE special register.
Name of sort control statement dataset; the default is IGZSRTCD.	SORT-CONTROL special register.
Name of sort message file; the default is SYSOUT.	SORT-MESSAGE special register.
Number of sort records.	SORT-FILE-SIZE special register.
Sort completion code.	SORT-RETURN special register.
Century window for sorting or merging on date fields.	YEARWINDOW compiler option.

The format of windowed date fields used as sort or merge keys are derived from PICTURE, USAGE, and DATE FORMAT clauses.