Getting Started using ICETOOL

Created by:

Samuel Ryan Smith

IBM DFSORT L2 Support

Xxx xxx, 200x

Notices and Disclaimers

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This information could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) at any time without notice.

Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

The information provided in this document is distributed "AS IS" without any warranty, either express or implied. IBM EXPRESSLY DISCLAIMS any warranties of merchantability, fitness for a particular purpose OR NON INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Agenda

- What is ICETOOL?
- ICETOOL Utility Operators
- Generating Sample data
- Basic ICETOOL JCL
- The SORT operator
- Creating multiple identical copies
- Using ICETOOL to collect statistics
- The STATS Operator/statistical output
- Counting values in a range
- Creating Reports
- Edit Masks, Leading Zeros, Edit Patterns, No Stats, and Division
- Leading, Floating and Trailing Characters
- Printing Sectioned Reports

What is ICETOOL?

- ICETOOL is a batch front-end utility that uses the capabilities of DFSORT to perform the operations you request.

- ICETOOL includes thirteen operators that perform sort, copy, statistical, report, selection, and splice operations.

- ICETOOL automatically calls DFSORT with the particular DFSORT control statements and options required for each operation (such as DYNALLOC for sorting).

- ICETOOL also produces messages and return codes describing the results of each operation and any errors detected. Although you generally do not need to look at the DFSORT messages produced as a result of an ICETOOL run, they are available in a separate data set if you need them.

- ICETOOL can be called directly or from a program. ICETOOL allows operator statements (and comments) to be supplied in a data set or in a parameter list passed by a calling program. For each operator supplied in the parameter list, ICETOOL puts information in the parameter list pertaining to that operation, thus allowing the calling program to use the information derived by ICETOOL.

Using the ICETOOL Utility, Operators

- **COPY** Copies a data set to one or more output data sets.
- **COUNT** Prints a message containing the count of records in a data set.
- **DEFAULTS** Prints the DFSORT installation defaults in a separate list data set.
- **DISPLAY** Prints the values or characters of specified numeric or character fields in a separate list data set. Simple, tailored, or sectioned reports can be produced.
- **MODE** Three modes are available which can be set or reset for groups of operators:
  - **STOP** mode (the default) stops subsequent operations if an error is detected.
  - **CONTINUE** mode continues with subsequent operations if an error is detected.
  - **SCAN** mode allows ICETOOL statement checking without actually performing any operations.
- **OCCUR** Prints each unique value for specified numeric or character fields and how many times it occurs in a separate list data set. Simple or tailored reports can be produced. The values printed can be limited to those for which the value count meets specified criteria (for example, only duplicate values or only non-duplicate values).
- **RANGE** Prints a message containing the count of values in a specified range for a specified numeric field in a data set.
- **SELECT** Selects records from a data set for inclusion in an output data set based on meeting criteria for the number of times specified numeric or character field values occur (for example, only duplicate values or only non-duplicate values).
- **SORT** Sorts a data set to one or more output data sets.
- **STATS** Prints messages containing the minimum, maximum, average, and total for specified numeric fields in a data set.
- **UNIQUE** Prints a message containing the count of unique values for a specified numeric or character field.
- **VERIFY** Examines specified decimal fields in a data set and prints a message identifying each invalid value found for each field.

Generating your sample data

```
//ICEDATA  JOB <JOB PARAMETERS>
//*
//****************************************************************
//*                                                              *
//* ICEDATA - CREATES THE SAMPLE FILES THAT MAY BE USED FOR      *
//*           INSTALLATION VERIFICATION AND FOR THE EXERCISES    *
//*           IN THE "GETTING STARTED WITH DFSORT" PUBLICATION.  *
//*                                                              *
//*    PROPRIETARY V3 STATEMENT                                  *
//*    LICENSED MATERIALS - PROPERTY OF IBM                      *
//*    5694-A01                                                  *
//*    (C) COPYRIGHT IBM CORP. 2003                              *
//*    END PROPRIETARY V3 STATEMENT                              *
//*                                                              *
//****************************************************************
//*                                                              *
//* INSTRUCTIONS:                                                *
//*   1. CHANGE THE JOB CARD TO MEET YOUR SYSTEM REQUIREMENTS.   *
//*   2. IN SAMPCOPY STEP BELOW, UNCOMMENT THE STEPLIB DD IF     *
//*      DFSORT'S LIBRARIES ARE NOT IN THE LINK LIST OR NOT IN   *
//*      THE LPA LIST. CHANGE HHH TO THE HIGH LEVEL QUALIFIER    *
//*      USED TO INSTALL DFSORT.                                 *
//*   3. CHANGE THE HIGH LEVEL QUALIFIER ON SAMPLE DATA SETS     *
//*      FROM HLQ TO ALLOWABLE HIGH LEVEL QUALIFIER FOR YOUR     *
//*      SYSTEM.                                                 *
//*                                                              *
//* RETURN CODE 0 IS EXPECTED.                                   *
//*                                                              *
//****************************************************************
//*
//SAMPCOPY EXEC PGM=ICESAMP,PARM=(BOOKS)
//*STEPLIB  DD  DSN=HHH.SICELINK,DISP=SHR
```

Generating your sample data

```
//*         DD  DSN=HHH.SORTLPA,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SAMPLE   DD   DSN=YOURHLQ.SORT.SAMPIN,DISP=(NEW,CATLG),
//         SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=173,BLKSIZE=1730),
//         UNIT=SYSALLDA
//ADD      DD   DSN=YOURHLQ.SORT.SAMPADD,DISP=(NEW,CATLG),
//         SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=173,BLKSIZE=1730),
//         UNIT=SYSALLDA
//OUTPUT   DD   DSN=YOURHLQ.SORT.SAMPOUT,DISP=(NEW,CATLG),
//         SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=173,BLKSIZE=1730),
//         UNIT=SYSALLDA
//BRANCH   DD   DSN=YOURHLQ.SORT.BRANCH,DISP=(NEW,CATLG),
//         SPACE=(TRK,(1,1),RLSE),
//         DCB=(RECFM=FB,LRECL=33,BLKSIZE=330),
//         UNIT=SYSALLDA
//SYSIN    DD   DUMMY
//*    =====> END OF JOB ICEDATA <=====
```

- This JOB will create 4 sample datasets that can be used for the examples in this presentation. The **ICEDATA** JOB can also be located in **SYS1.SICESAMP**.
  - All of the sample JOBs in this presentation will be provided in a PDS as well.
  - DFSORT TOOLKIT - **L2.SAMPLE.SORT.EXAMPLE.JOBS.TOOLKIT**

ICETOOL required JCL statements

An ICETOOL job consists of:

- The JCL statements that are required for every ICETOOL job.
- The operator statements indicating the operations to be performed by the ICETOOL job.
- The JCL statements that are required as a result of the specified operator statements.

Here is a skeleton of the JCL for an ICETOOL JOB:

```
//EXAMP    JOB   A492,PROGRAMMER
//TOOL     EXEC  PGM=ICETOOL
//TOOLMSG  DD    SYSOUT=A
//DFSMSG   DD    SYSOUT=A
//TOOLIN   DD    *
  <ICETOOL statements go here>
/*
<Additional JCL statements go here>
```

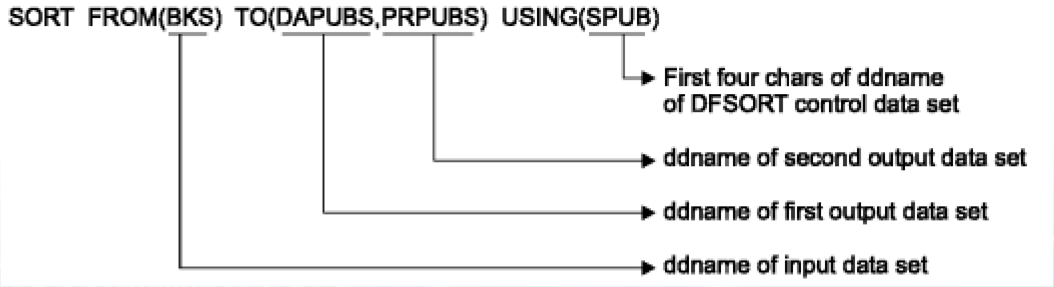
Steps for writing the SORT operator

1. Write a comment statement (optional):* Books from VALD and WETH*
2. Type **SORT** after the comment statement
3. Leave at least one blank and type FROM(BKS) BKS specifies the ddname for the input data sets you want to sort.
4. Leave at least one blank and type TO(DAPUBS,PRPUBS) TO specifies the ddnames for the output data sets to contain the sorted subset of records. You can create up to 10 identical output data sets of any type that DFSORT allows (permanent, temporary, disk, tape, print, etc). In this case, DAPUBS is the ddname chosen for the temporary disk data set and PRPUBS is the ddname chosen for the print data set. You can use any valid 1-8 character ddnames you like.
5. Leave at least one blank and type USING(SPUB) USING specifies the first four characters of the ddname for the data set containing the DFSORT control statements. In this case, the four characters chosen are SPUB, but you can use any four characters you like as long as they are valid for a ddname. The last four characters of the ddname are always CNTL, so in this case the full ddname is SPUBCNTL. For the SORT operator, you must specify a SORT control statement in the DFSORT control statement data set (SPUBCNTL) in order to tell DFSORT how to sort the input data set. You can also specify additional DFSORT control statements, like INCLUDE, OMIT, INREC, OUTREC and OUTFIL, as appropriate.

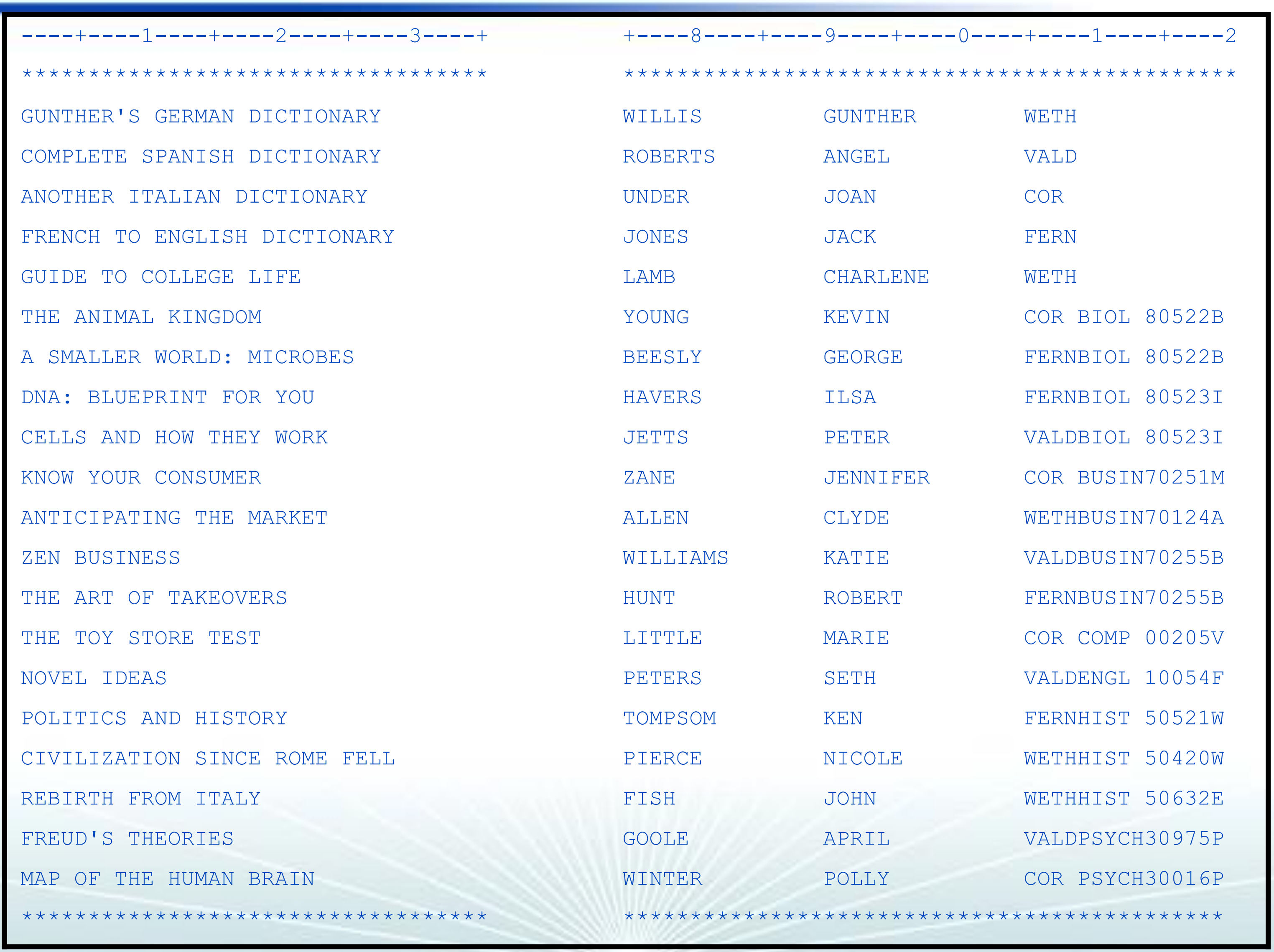
Using the ICETOOL Utility, creating multiple identical copies

```
//TOOL          EXEC  PGM=ICETOOL
//TOOLMSG  DD SYSOUT=*
//DFSMSG   DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//TOOLIN   DD *
* BOOKS FROM VALD AND WETH
   SORT FROM(BKS) TO(DAPUBS,PRPUBS) USING(SPUB)
/*
//BKS      DD DSN=S744428.SORT.SAMPIN,DISP=SHR
//         DD DSN=S744428.SORT.SAMPADD,DISP=SHR
//DAPUBS   DD DSN=L2.SAMPLE.SORT.DAPUBS,
//         DISP=(NEW,CATLG),SPACE=(CYL,(5,5)),UNIT=SYSDA
//PRPUBS   DD DSN=L2.SAMPLE.SORT.PRPUBS,
//         DISP=(NEW,CATLG),SPACE=(CYL,(5,5)),UNIT=SYSDA
//SPUBCNTL DD *
   SORT FIELDS=(106,4,A,1,75,A),FORMAT=CH
    INCLUDE COND=(106,4,EQ,C'VALD',OR,106,4,EQ,C'WETH'),
            FORMAT=CH
/*
```

- L2.SAMPLE.SORT.ICETOOL.JOBLOG

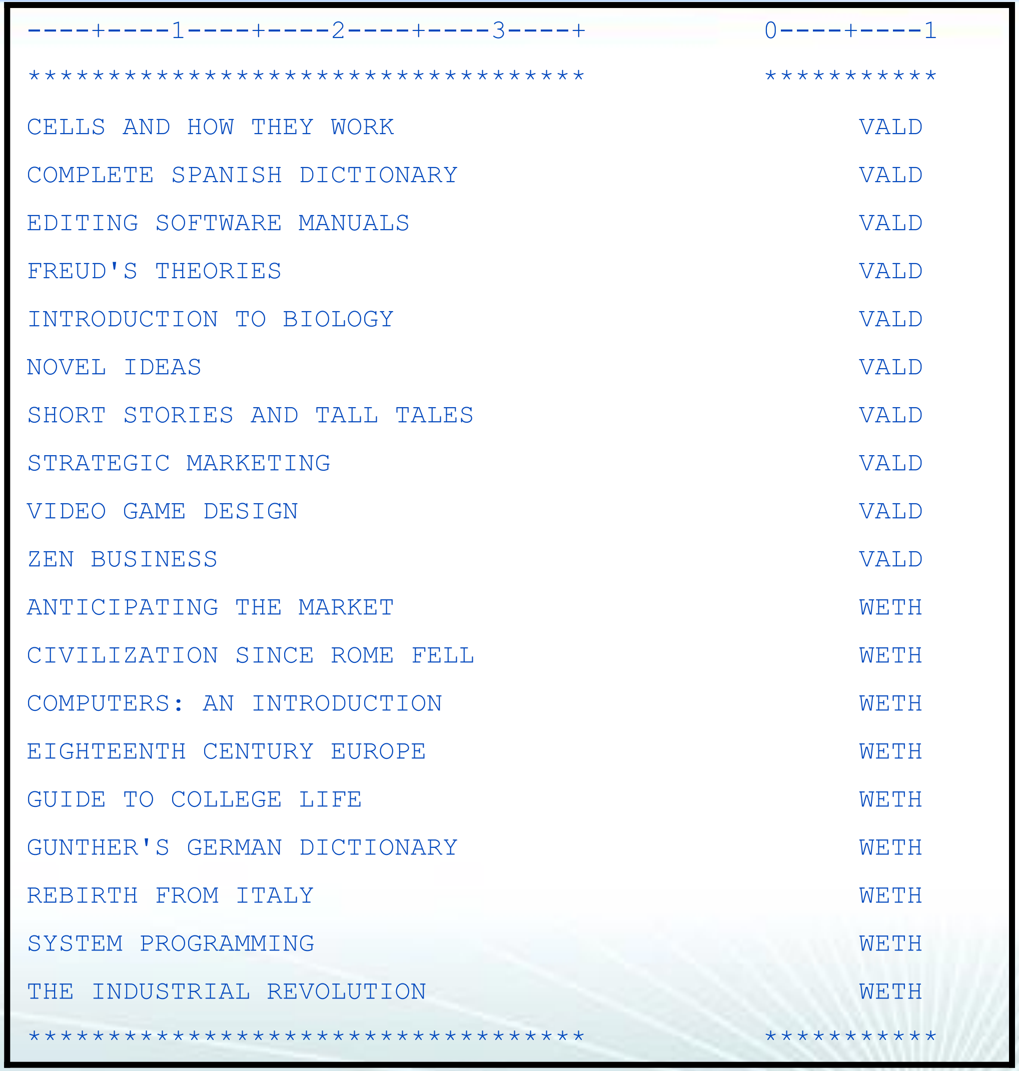

SORT FROM(BKS) TO(DAPUBS,PRPUBS) USING(SPUB)
First four chars of ddname of DFSORT control data set
ddname of second output data set
ddname of first output data set
ddname of input data set

Data prior to running ICETOOL, from dataset SAMPIN



```
----+----1----+----2----+----3----+          +----8----+----9----+----0----+----1----+----2
***********************************          **********************************************
GUNTHER'S GERMAN DICTIONARY                  WILLIS         GUNTHER        WETH
COMPLETE SPANISH DICTIONARY                  ROBERTS        ANGEL          VALD
ANOTHER ITALIAN DICTIONARY                   UNDER          JOAN           COR
FRENCH TO ENGLISH DICTIONARY                 JONES          JACK           FERN
GUIDE TO COLLEGE LIFE                        LAMB           CHARLENE       WETH
THE ANIMAL KINGDOM                           YOUNG          KEVIN          COR BIOL 80522B
A SMALLER WORLD: MICROBES                    BEESLY         GEORGE         FERNBIOL 80522B
DNA: BLUEPRINT FOR YOU                       HAVERS         ILSA           FERNBIOL 80523I
CELLS AND HOW THEY WORK                      JETTS          PETER          VALDBIOL 80523I
KNOW YOUR CONSUMER                           ZANE           JENNIFER       COR BUSIN70251M
ANTICIPATING THE MARKET                      ALLEN          CLYDE          WETHBUSIN70124A
ZEN BUSINESS                                 WILLIAMS       KATIE          VALDBUSIN70255B
THE ART OF TAKEOVERS                         HUNT           ROBERT         FERNBUSIN70255B
THE TOY STORE TEST                           LITTLE         MARIE          COR COMP 00205V
NOVEL IDEAS                                  PETERS         SETH           VALDENGL 10054F
POLITICS AND HISTORY                         TOMPSOM        KEN            FERNHIST 50521W
CIVILIZATION SINCE ROME FELL                 PIERCE         NICOLE         WETHHIST 50420W
REBIRTH FROM ITALY                           FISH           JOHN           WETHHIST 50632E
FREUD'S THEORIES                             GOOLE          APRIL          VALDPSYCH30975P
MAP OF THE HUMAN BRAIN                       WINTER         POLLY          COR PSYCH30016P
***********************************          *********************************************
```

Output from ICETOOL, in datasets PRPUBS and DAPUBS



```
----+----1----+----2----+----3----+           0----+----1
***********************************           ***********
CELLS AND HOW THEY WORK                             VALD
COMPLETE SPANISH DICTIONARY                         VALD
EDITING SOFTWARE MANUALS                            VALD
FREUD'S THEORIES                                    VALD
INTRODUCTION TO BIOLOGY                             VALD
NOVEL IDEAS                                         VALD
SHORT STORIES AND TALL TALES                        VALD
STRATEGIC MARKETING                                 VALD
VIDEO GAME DESIGN                                   VALD
ZEN BUSINESS                                        VALD
ANTICIPATING THE MARKET                             WETH
CIVILIZATION SINCE ROME FELL                        WETH
COMPUTERS: AN INTRODUCTION                          WETH
EIGHTEENTH CENTURY EUROPE                           WETH
GUIDE TO COLLEGE LIFE                               WETH
GUNTHER'S GERMAN DICTIONARY                         WETH
REBIRTH FROM ITALY                                  WETH
SYSTEM PROGRAMMING                                  WETH
THE INDUSTRIAL REVOLUTION                           WETH
***********************************           ***********
```

```
SORT FIELDS=(106,4,A,1,75,A),FORMAT=CH
INCLUDECOND=(106,4,EQ,C'VALD',OR,106,4,EQ,C'WETH'),
            FORMAT=CH
```

Using ICETOOL to collect statistics

```
//TOOLIN   DD *
* STATISTICS FROM ALL BRANCHES
     STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
* BOOKS FROM VALD AND WETH
     SORT FROM(BKS) TO(DAPUBS,PRPUBS) USING(SPUB)
/*
//ALL      DD DSN=S744428.SORT.BRANCH,DISP=SHR
//BKS      DD DSN=S744428.SORT.SAMPIN,DISP=SHR
//         DD DSN=S744428.SORT.SAMPADD,DISP=SHR
//DAPUBS   DD DSN=L2.SAMPLE.SORT.DAPUBS2,
//            DISP=(NEW,CATLG),SPACE=(CYL,(5,5)),UNIT=SYSDA
//PRPUBS   DD DSN=L2.SAMPLE.SORT.PRPUBS2,
//            DISP=(NEW,CATLG),SPACE=(CYL,(5,5)),UNIT=SYSDA
//SPUBCNTL DD *
   SORT FIELDS=(106,4,A,1,75,A),FORMAT=CH
    INCLUDE COND=(106,4,EQ,C'VALD',OR,106,4,EQ,C'WETH'),
            FORMAT=CH
/*
```

- JOBLOG can be located on SNJMAS3 →**L2.SAMPLE.SORT.ICETOOL2.JOBLOG**
  - Added 3 lines in **RED**

Steps to create a STATS operator

1. Type STATS after the comment statement (you can leave one or more blanks before STATS if you like).
2. Leave at least one blank and type FROM(ALL) FROM specifies the ddname (that is, the name of the DD statement) for the input data set from which you want to print statistics. In this case ALL is the ddname chosen, but you can use any valid 1-8 character ddname you like.
3. Leave at least one blank and type ON. ON defines a field for which you want to print statistics.
4. Type in parentheses, and separated by commas:
   1. Where the employees field begins relative to the beginning of the input record (the first position is byte The employees field begins at byte 18.
   2. The length of the employees field in bytes. The employees field is 4 bytes long.
   3. A code for the data format. The employees field contains zoned decimal data, which you specify as ZD.
5. Leave at least one blank and type ON. ON defines another field for which you want to print statistics. You can print statistics for up to 10 fields with one STATS statement. Specify the ON fields in the same order in which you want their statistics to be printed.
6. Type in parentheses, and separated by commas the location (28), length (6), and format (PD for packed decimal) of the profit field.
7. Leave at least one blank and type ON. Type in parentheses and separated by commas, the location (22), length (6), and format (PD) of the revenue field. Make sure that the statement is coded between columns 1 and 72.

Continuing an Operator Statement

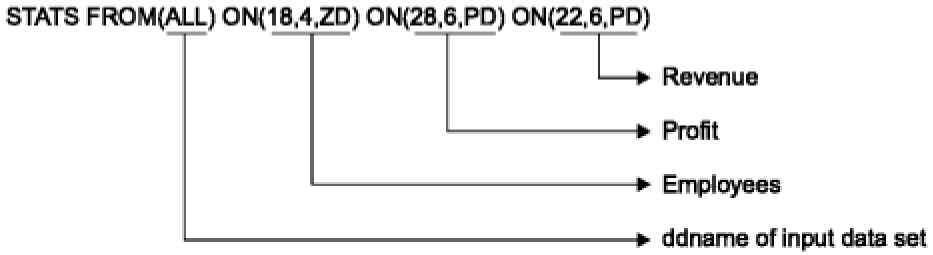
- If you cannot fit your STATS statement (or any other ICETOOL operator statement) between columns 1 and 72 of a single line, you can continue it across multiple lines. If you end a line with a hyphen (-) after the operator or any operand, the next line is treated as a continuation. Any characters specified after the hyphen are ignored.

- Note that the operator and each operand must be completely specified on one line (between columns 1 and 72).

Example

```
STATS - this is the operator
      FROM(ALL) - ALL is the ddname for SORT.BRANCH
            ON(18,4,ZD)-
            ON(28,6,PD)-
            ON(22,6,PD)
```

Data from the BRANCH dataset



```
STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
```

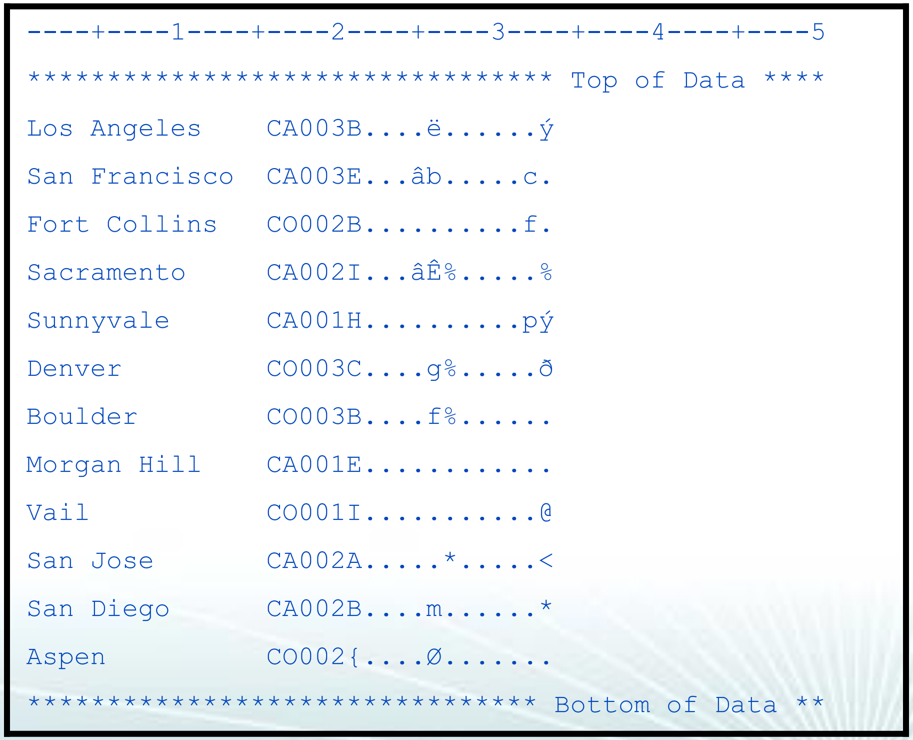
Revenue

Profit

Employees

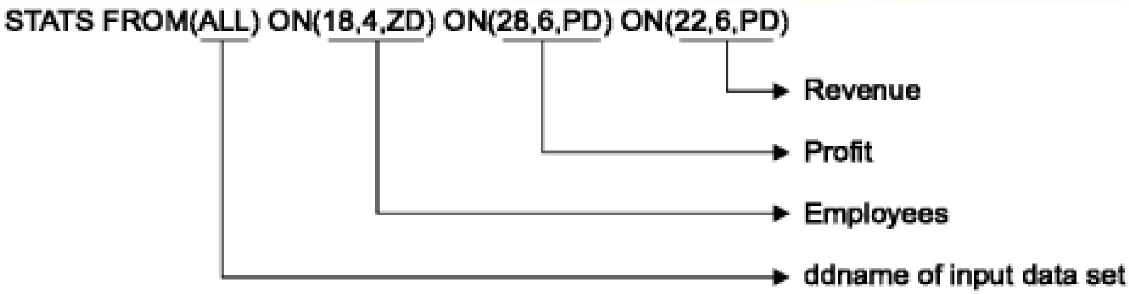
ddname of input data set

- Note that the data in columns 22 through 34 is not readable since these values are currently zoned and packed decimals.

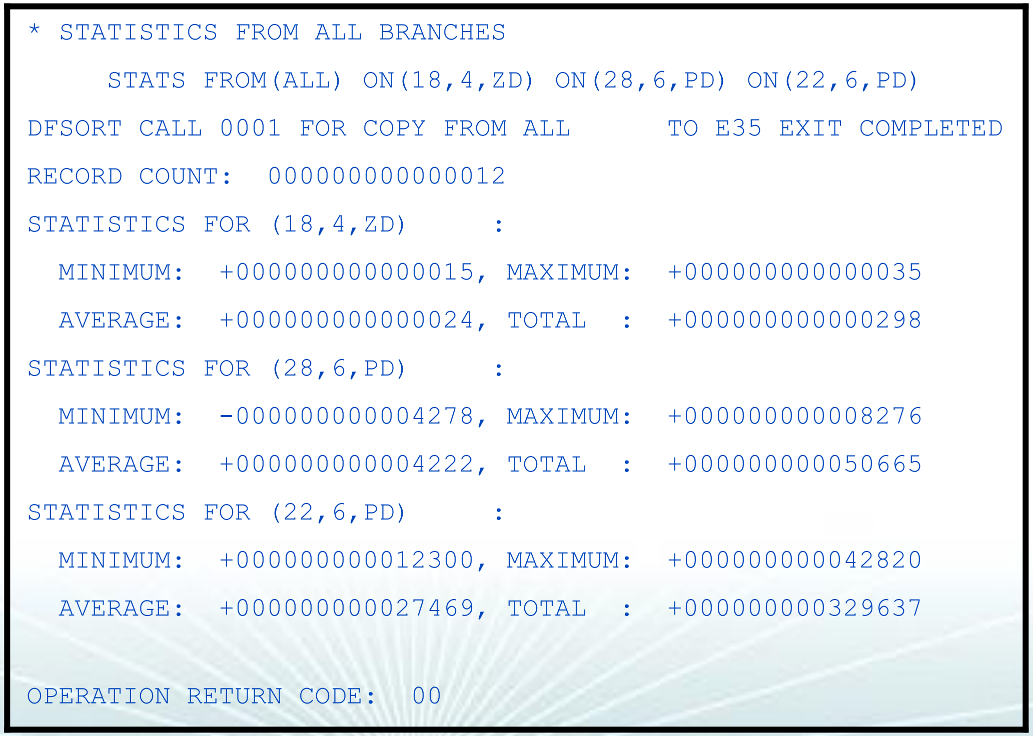


```
----+----1----+----2----+----3----+----4----+----5
********************************* Top of Data ****
Los Angeles    CA003B....ë......ý
San Francisco  CA003E...âb.....c.
Fort Collins   CO002B..........f.
Sacramento     CA002I...âÊ%.....%
Sunnyvale      CA001H..........pý
Denver         CO003C....g%.....ð
Boulder        CO003B....f%......
Morgan Hill    CA001E............
Vail           CO001I...........@
San Jose       CA002A.....*.....<
San Diego      CA002B....m......*
Aspen          CO002{....Ø.......
******************************** Bottom of Data **
```

Output for statistics from ICETOOL



STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)

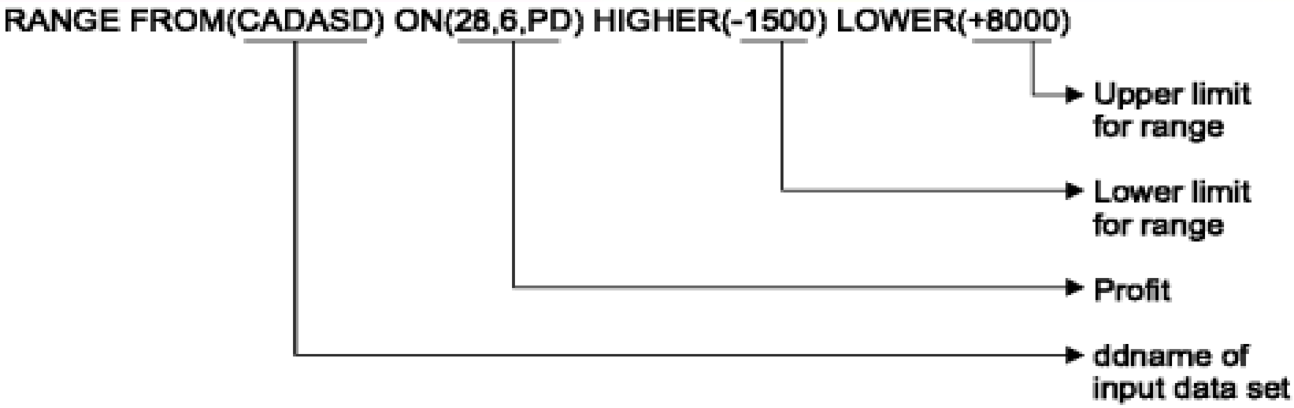


```
* STATISTICS FROM ALL BRANCHES
     STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
DFSORT CALL 0001 FOR COPY FROM ALL      TO E35 EXIT COMPLETED
RECORD COUNT:  000000000000012
STATISTICS FOR (18,4,ZD)      :
  MINIMUM:  +000000000000015, MAXIMUM:  +000000000000035
  AVERAGE:  +000000000000024, TOTAL   :  +000000000000298
STATISTICS FOR (28,6,PD)      :
  MINIMUM:  -000000000004278, MAXIMUM:  +000000000008276
  AVERAGE:  +000000000004222, TOTAL   :  +000000000050665
STATISTICS FOR (22,6,PD)      :
  MINIMUM:  +000000000012300, MAXIMUM:  +000000000042820
  AVERAGE:  +000000000027469, TOTAL   :  +000000000329637

OPERATION RETURN CODE:  00
```

Counting Values in a Range

```
//TOOLIN   DD *
* SEPARATE OUTPUT FOR CALIFORNIA AND COLORADO BRANCHES
    SORT FROM(ALL) USING(CACO)
* CALIFORNIA BRANCHES PROFIT ANALYSIS
    RANGE FROM(CADASD) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)
* BRANCHES WITH LESS THAN 32 EMPLOYEES
    RANGE FROM(ALL) ON(18,4,ZD) LOWER(32)
/*
//ALL    DD DSN=S744428.SORT.BRANCH,DISP=SHR
/*
//CACOCNTL DD *
    SORT FIELDS=(1,15,CH,A)
    OUTFIL FNAMES=(CADASD,CATAPE),INCLUDE=(16,2,CH,EQ,C'CA')
    OUTFIL FNAMES=(CODASD,COTAPE),INCLUDE=(16,2,CH,EQ,C'CO')
/*
//CADASD DD  DSN=&&CA,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//CATAPE DD DSN=CA.BRANCH,UNIT=VTAPE,
//  DISP=(NEW,KEEP)
//CODASD DD  DSN=&&CO,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//COTAPE DD DSN=CO.BRANCH,UNIT=VTAPE,
//  DISP=(NEW,KEEP)
//OUT DD SYSOUT=*
//RPT DD SYSOUT=*
//SECTIONS DD SYSOUT=*
//BKIN DD DSN=S744428.SORT.SAMPIN,DISP=SHR
//PUBCT DD SYSOUT=*
//BKOUT DD DSN=S744428.BOOKS1,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(3,3)),UNIT=3390
/*
```

Output for Counting Values in a Range



RANGE FROM(CADASD) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)

Upper limit for range

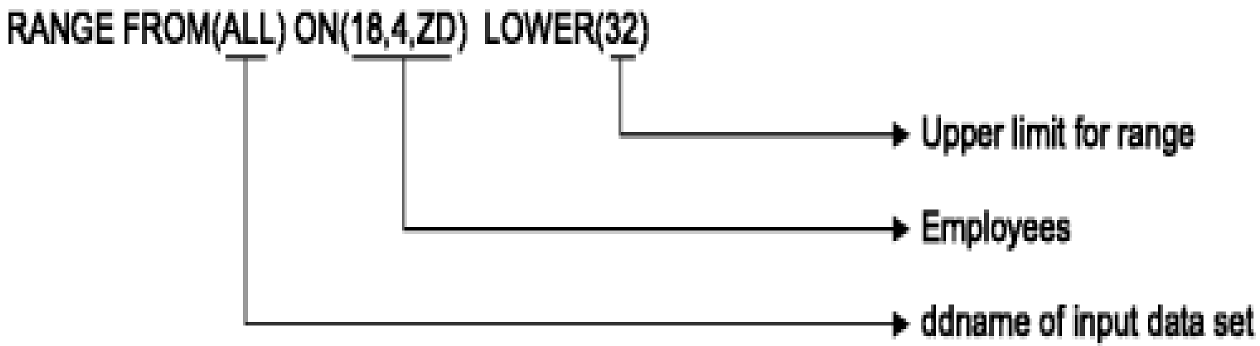
Lower limit for range

Profit

ddname of input data set

```
          * CALIFORNIA BRANCHES PROFIT ANALYSIS
              RANGE FROM(CADASD) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)
ICE627I 0 DFSORT CALL 0002 FOR COPY FROM CADASD   TO E35 EXITCOMPLETED
ICE628I 0 RECORD COUNT:  000000000000007
ICE631I 0 NUMBER OF VALUES IN RANGE FOR (28,6,PD)     :  000000000000003

ICE602I 0 OPERATION RETURN CODE:  00
```

Output for Counting Values in a Range, cont’d


RANGE FROM(ALL) ON(18,4,ZD) LOWER(32)

Upper limit for range

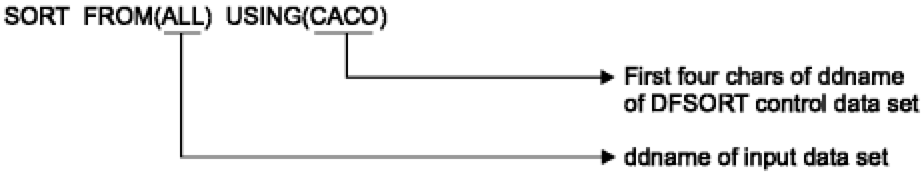
Employees

ddname of input data set

```
       * BRANCHES WITH LESS THAN 32 EMPLOYEES
               RANGE FROM(ALL) ON(18,4,ZD) LOWER(32)
ICE627I 0 DFSORT CALL 0003 FOR COPY FROM ALL      TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT:  000000000000012
ICE631I 0 NUMBER OF VALUES IN RANGE FOR (18,4,ZD)     :   000000000000008
ICE602I 0 OPERATION RETURN CODE:  00
```

Creating Different Subsets of a Sorted Data Set and/or Tailored Reports

```
//TOOLIN   DD *
* SEPARATE OUTPUT FOR CALIFORNIA AND COLORADO BRANCHES
     SORT FROM(ALL) USING(CACO)
* PRINT A REPORT FOR THE COLORADO BRANCHES
   DISPLAY FROM(CODASD) LIST(RPT) -
       DATE TITLE('COLORADO BRANCHES REPORT') PAGE -
       HEADER('CITY') HEADER('PROFIT') HEADER('EMPLOYEES') -
       ON(1,15,CH) ON(28,6,PD) ON(18,4,ZD) BLANK -
       TOTAL('TOTAL') AVERAGE('AVERAGE') MINIMUM('LOWEST')
/*
//ALL     DD DSN=S744428.SORT.BRANCH,DISP=SHR
/*
//CACOCNTL DD *
     SORT FIELDS=(1,15,CH,A)
     OUTFIL FNAMES=(CADASD,CATAPE),INCLUDE=(16,2,CH,EQ,C'CA')
     OUTFIL FNAMES=(CODASD,COTAPE),INCLUDE=(16,2,CH,EQ,C'CO')
/*
//CADASD DD  DSN=&&CA,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//CATAPE DD DSN=CA.BRANCH,UNIT=VTAPE,
//  DISP=(NEW,KEEP)
//CODASD DD  DSN=&&CO,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//COTAPE DD DSN=CO.BRANCH,UNIT=VTAPE,
//  DISP=(NEW,KEEP)
//OUT DD SYSOUT=*
//RPT DD SYSOUT=*
//SECTIONS DD SYSOUT=*
//BKIN DD DSN=S744428.SORT.SAMPIN,DISP=SHR
//PUBCT DD SYSOUT=*
//BKOUT DD DSN=S744428.BOOKS1,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(3,3)),UNIT=3390
```

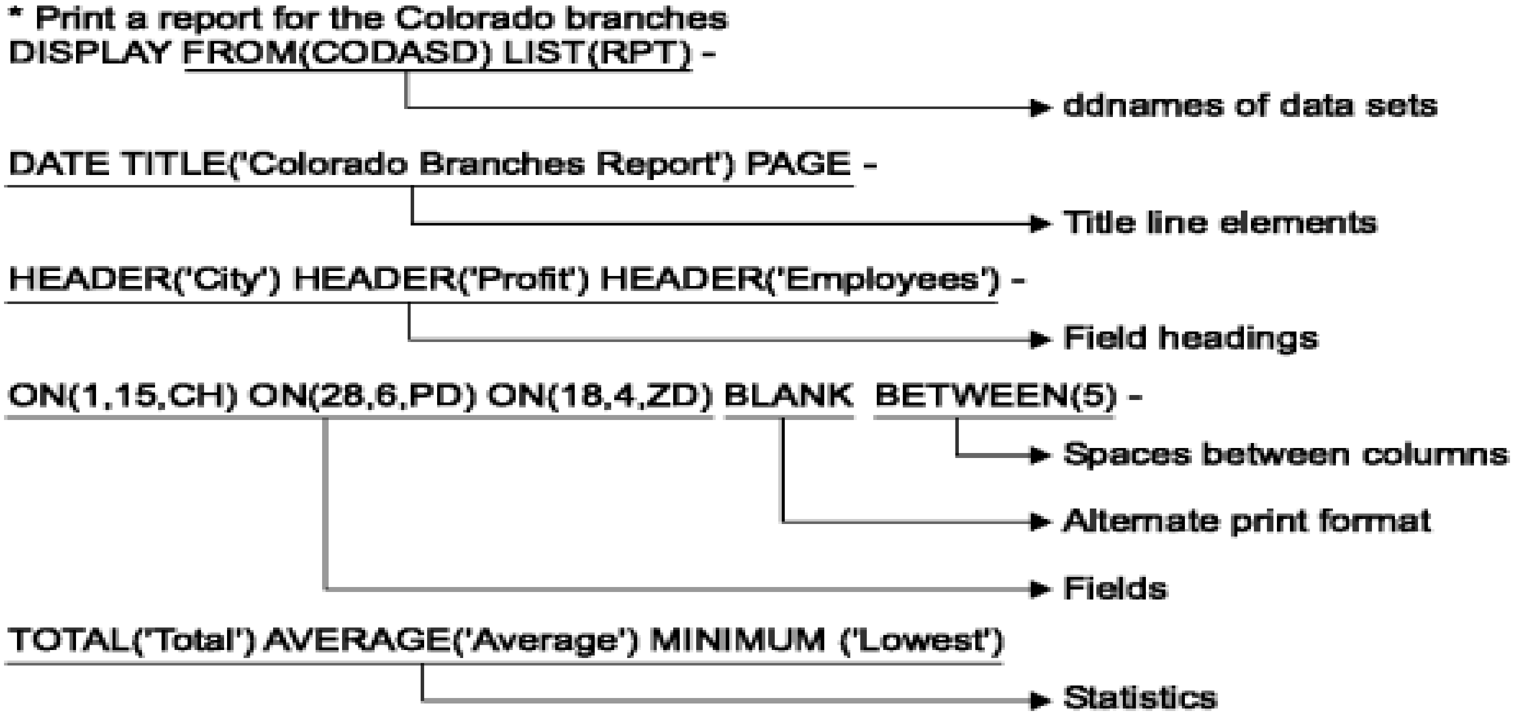


SORT FROM(ALL) USING(CACO)

First four chars of ddname of DFSORT control data set

ddname of input data set

Creating Different Subsets of a Sorted Data Set and/or Tailored Reports

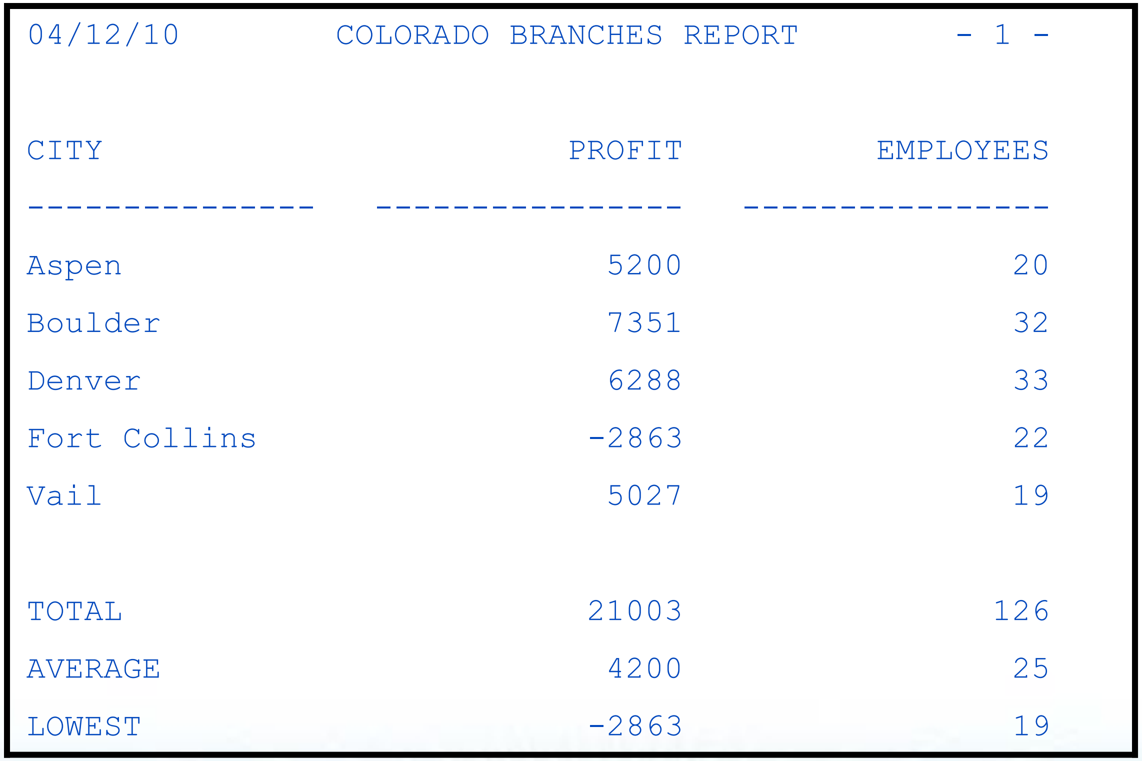


```
* Print a report for the Colorado branches
DISPLAY FROM(CODASD) LIST(RPT) -
DATE TITLE('Colorado Branches Report') PAGE -
HEADER('City') HEADER('Profit') HEADER('Employees') -
ON(1,15,CH) ON(28,6,PD) ON(18,4,ZD) BLANK  BETWEEN(5) -
TOTAL('Total') AVERAGE('Average') MINIMUM ('Lowest')
```

- This is what we saw in the RED control statements from the previous slide

Output from different subsets and/or Tailored reports

- JOBLOG can be located on SNJMAS3 →**L2.SAMPLE.SORT.ICETOOL.JLOG.CACO**



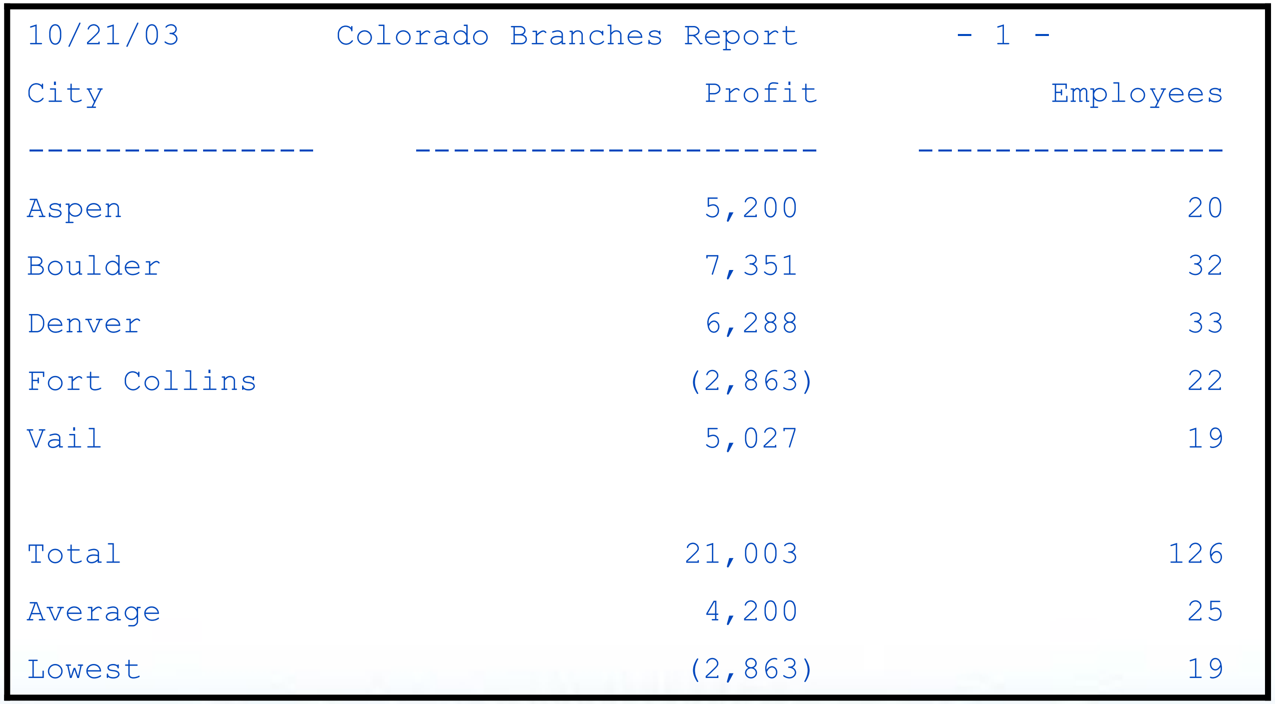
```
04/12/10        COLORADO BRANCHES REPORT        - 1 -


CITY                          PROFIT          EMPLOYEES

---------------   ----------------   ----------------

Aspen                           5200                 20

Boulder                         7351                 32

Denver                          6288                 33

Fort Collins                   -2863                 22

Vail                            5027                 19


TOTAL                          21003                126

AVERAGE                         4200                 25

LOWEST                         -2863                 19
```

Edit Masks

- Thirty-three pre-defined edit masks
  - **d** is used to represent a decimal digit (0-9)
  - **w** is used to represent a leading sign that will be blank for a positive value or - for a negative value
  - **x** is used to represent a trailing sign that will be blank for a positive value or - for a negative value
  - **y** is used to represent a leading sign that will be blank for a positive value or (for a negative value
  - **z** is used to represent a trailing sign that will be blank for a positive value or) for a negative value
- Edit Pattern Mask E1 would look like
  - yd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,dddz

Edit Masks

- Now If I simply add this Edit Pattern Mask changing ON(28,6,PD) to ON(28,6,PD,**E1**).



```
10/21/03        Colorado Branches Report        - 1 -
City                               Profit            Employees
---------------     ---------------------     ----------------
Aspen                              5,200                    20
Boulder                            7,351                    32
Denver                             6,288                    33
Fort Collins                      (2,863)                   22
Vail                               5,027                    19

Total                             21,003                   126
Average                            4,200                    25
Lowest                            (2,863)                   19
```

Leading Zeros

- By default, leading zeros are not displayed when you use an edit mask, but you can change that by adding LZ
  - HEADER('No leading zeros','(without LZ)') ON(28,6,PD,E1) –
  - HEADER('Leading zeros','(with LZ)') ON(28,6,PD,E1,**LZ**)

```
No leading zeros      Leading zeros
    (without LZ)          (with LZ)
----------------    ----------------
         (4,278)    (00,000,004,278)
           6,832     00,000,006,832
         (2,863)    (00,000,002,863)
           8,276     00,000,008,276
           (978)    (00,000,000,978)
           6,288     00,000,006,288
           7,351     00,000,007,351
           3,271     00,000,003,271
           5,027     00,000,005,027
           8,264     00,000,008,264
           8,275     00,000,008,275
           5,200     00,000,005,200
```

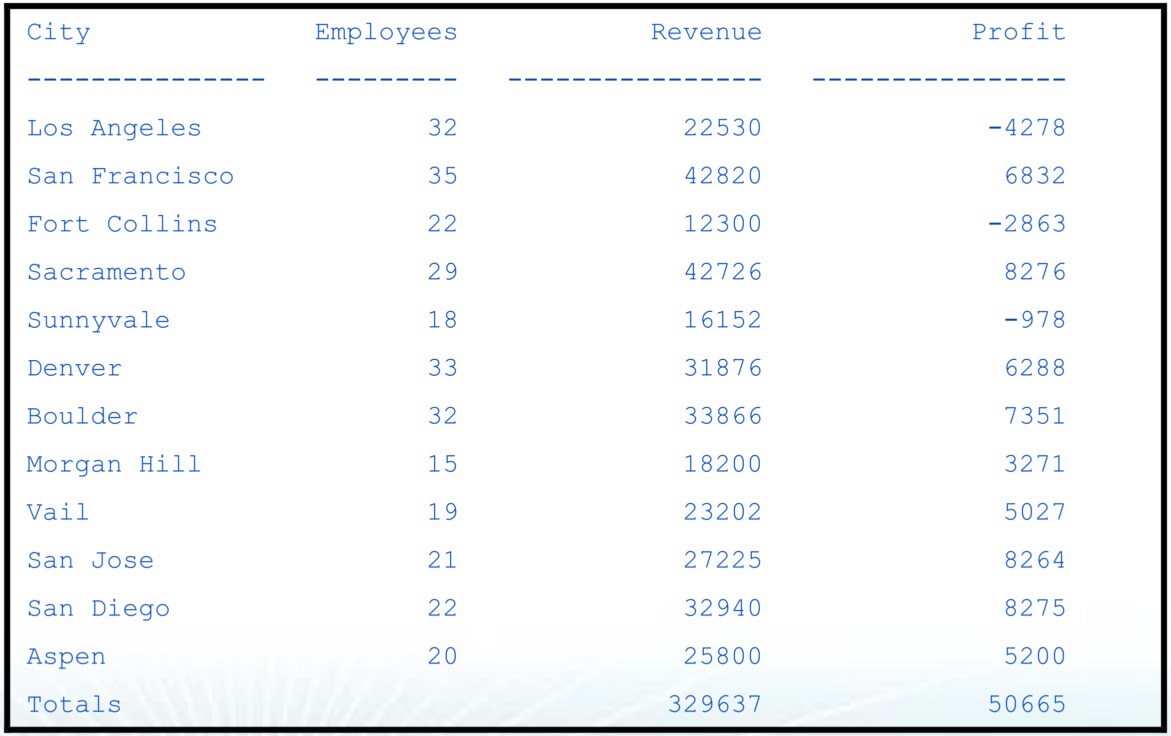
Edit Patterns

- Edit masks are not particularly useful for unsigned numeric data such as telephone numbers, dates, time-of-day, etc…
  - Instead use a 9 in the pattern where you want a digit (0-9) from the numeric value to appear
- For Example:
  - If you have an 8-byte ZD date in the form *mmddyyyy* in positions 21-28, you can display it as *mm/dd/yyyy* using ON(21,8,ZD,E'99/99/9999').
  - An 8-byte value of 03122004 is displayed as **03/12/2004**.
- Or:
  - If you have a 10-byte ZD telephone number in the form aaapppnnnn in positions 31-40, you can display it as (aaa)-ppp-nnnn using ON(31,10,ZD, E'(999)-999-9999').
  - A 10-byte value of 0123456789 is displayed as **(012)-345-6789**.

No Statistics

- By default, any statistics you request using TOTAL, MAXIMUM, MINIMUM, etc…. are displayed for every numeric ON field
- Use the **NOST** formatting item to suppress statistics
- Totals for the Revenue and Profit fields, but not for the Employees field

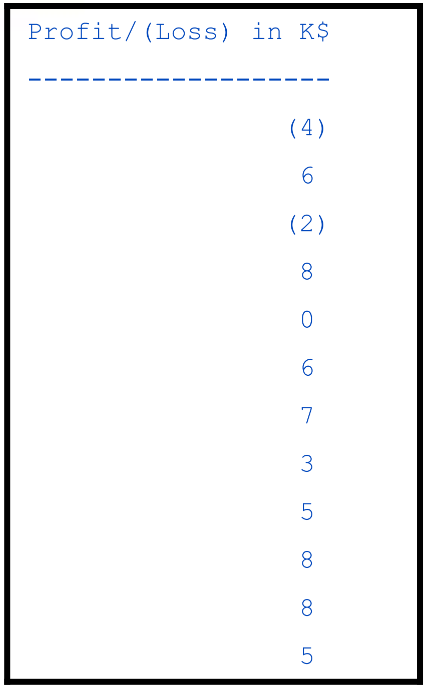
```
DISPLAY FROM(IN) LIST(RPT3) -
 HEADER('City') ON(1,15,CH) -
 HEADER('Employees') ON(18,4,ZD,NOST) -
 HEADER('Revenue') ON(22,6,PD) -
 HEADER('Profit') ON(28,6,PD) -
TOTAL('Totals')
```



```
City               Employees             Revenue              Profit
---------------    ---------    ----------------    ----------------

Los Angeles               32               22530               -4278
San Francisco             35               42820                6832
Fort Collins              22               12300               -2863
Sacramento                29               42726                8276
Sunnyvale                 18               16152                -978
Denver                    33               31876                6288
Boulder                   32               33866                7351
Morgan Hill               15               18200                3271
Vail                      19               23202                5027
San Jose                  21               27225                8264
San Diego                 22               32940                8275
Aspen                     20               25800                5200
Totals                                    329637               50665
```

Division

- Ten division items
  - **/D** - divide by 10
  - **/C** - divide by 100
  - **/K** - divide by 1000
  - **/DK** - divide by 10000 (10*1000)
  - **/CK** - divide by 100000 (100*1000)
  - **/M** - divide by 1000000 (1000*1000)
  - **/G** - divide by 1000000000 (1000*1000*1000)
  - **/KB** - divide by 1024
  - **/MB** - divide by 1048576 (1024*1024)
  - **/GB** - divide by 1073741824 (1024*1024*1024)



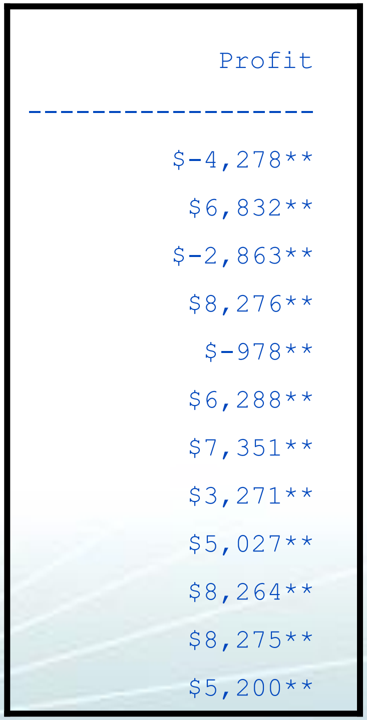
```
Profit/(Loss)  in K$
--------------------
                 (4)
                  6
                 (2)
                  8
                  0
                  6
                  7
                  3
                  5
                  8
                  8
                  5
```

- Using HEADER('Profit/(Loss) in K$') and ON(28,6,PD,E1,***/K***)

Leading, Floating and Trailing Characters

- You can add floating characters to your numeric fields and add leading and trailing characters to your numeric and character fields as follows:
  - **F'string'** - a floating string to appear to the left of the first non-blank character of the formatted numeric data
  - **L'string'** - a leading string to appear at the beginning of the character or numeric data column
  - **T'string'** - a trailing string to appear at the end of the character or numeric data column

- Using HEADER('Profit') and ON(28,6,PD,A1,**F'$',T'**'**)

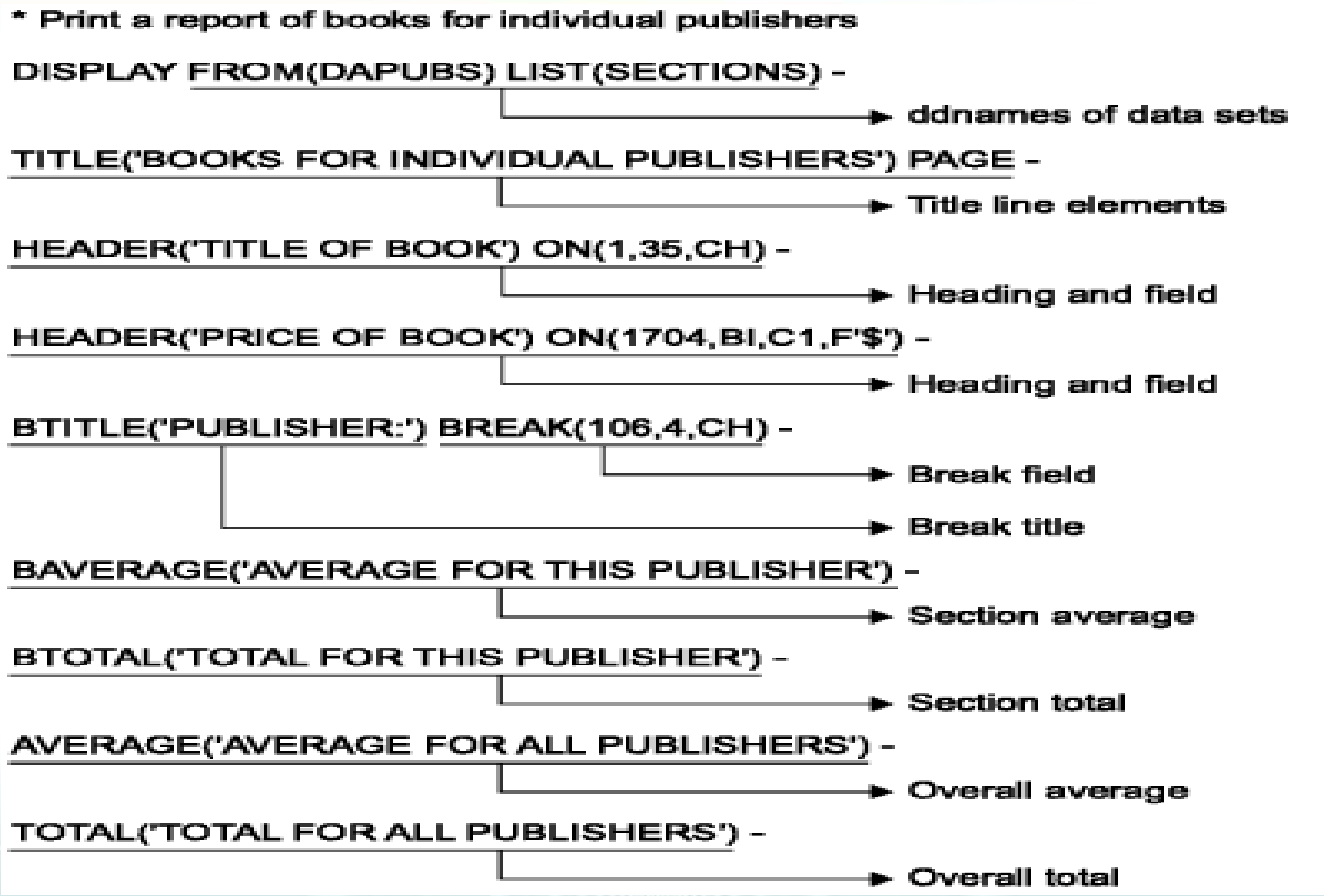


```
            Profit
------------------
         $-4,278**
          $6,832**
         $-2,863**
          $8,276**
           $-978**
          $6,288**
          $7,351**
          $3,271**
          $5,027**
          $8,264**
          $8,275**
          $5,200**
```

Printing Sectioned Reports

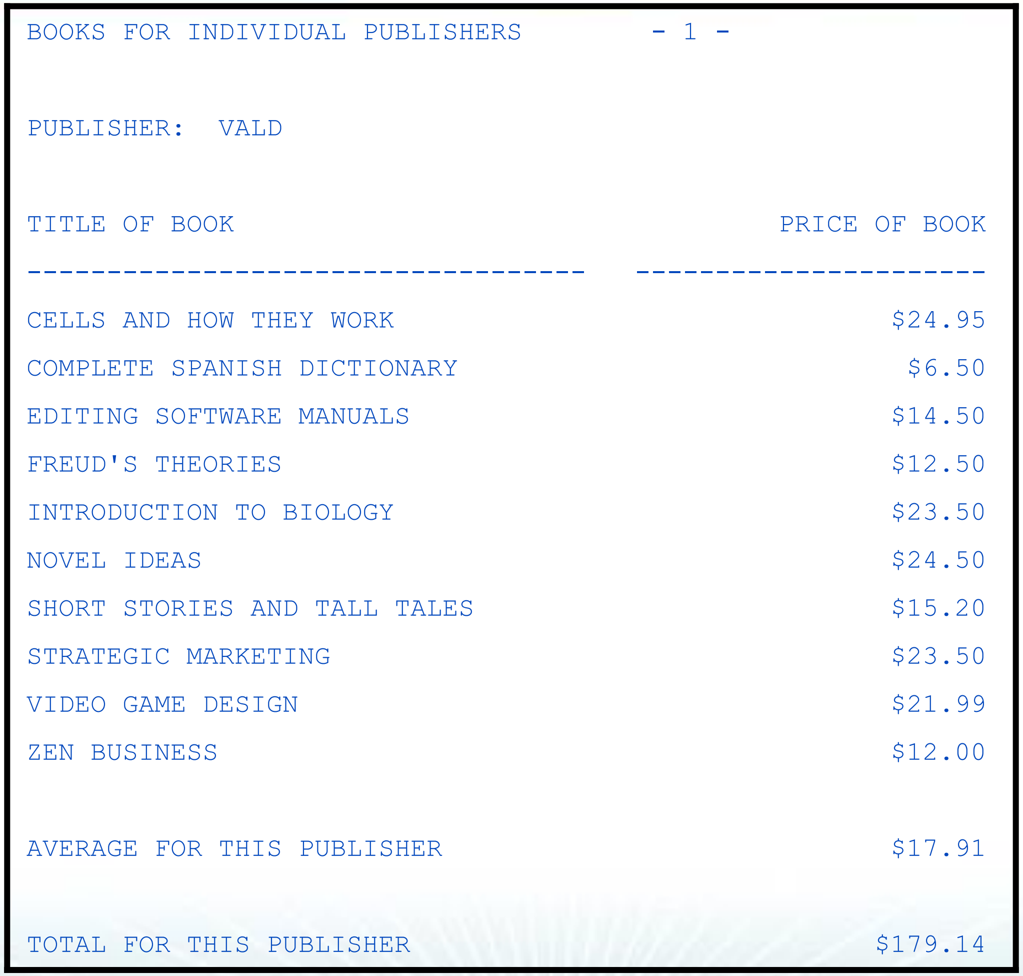
- Using the BREAK operand of DISPLAY, you can create reports divided into sections
- By a character or numeric break field on which you have previously sorted
- Format items with BREAK(p,m,f,formatting) in the same way you can use them with ON(p,m,f,formatting)
- Use break title (BTITLE operand) and statistics for the individual sections (BTOTAL, BAVERAGE, BMAXIMUM and BMINIMUM operands)

Printing Sectioned Reports



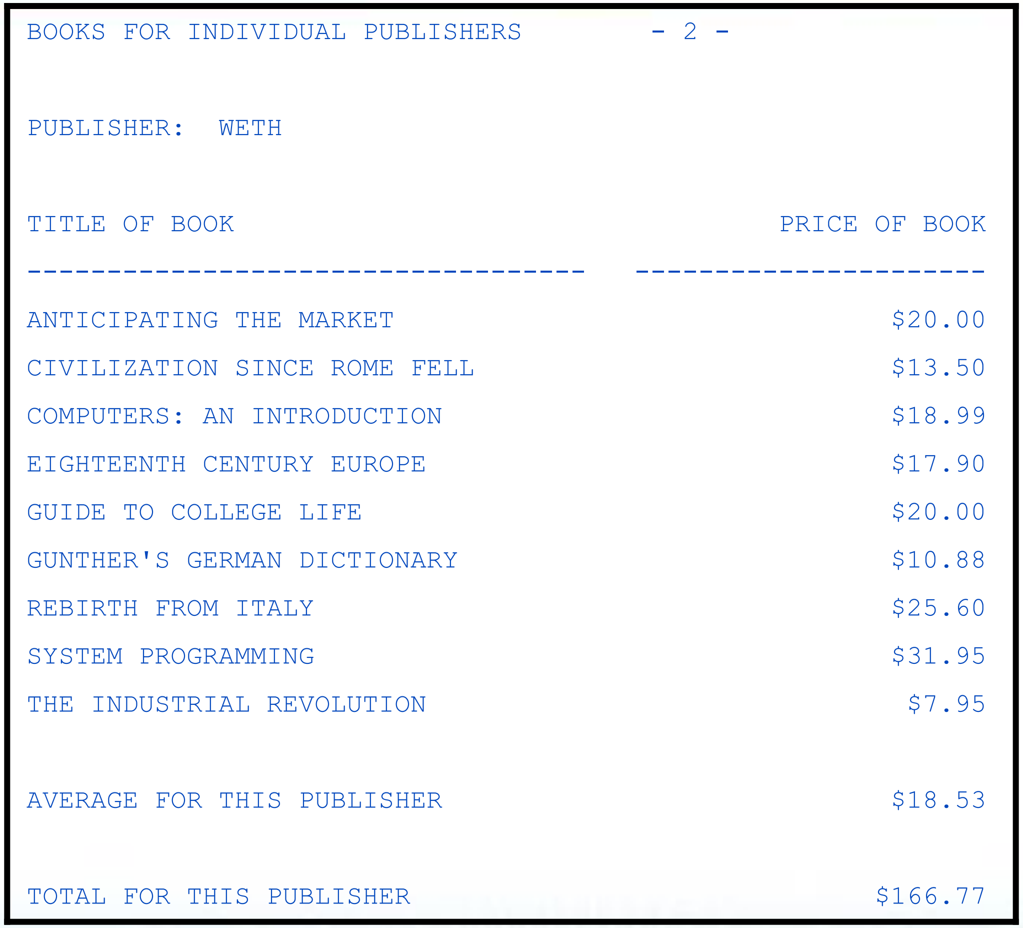
```
* Print a report of books for individual publishers
DISPLAY FROM(DAPUBS) LIST(SECTIONS) -
TITLE('BOOKS FOR INDIVIDUAL PUBLISHERS') PAGE -
HEADER('TITLE OF BOOK') ON(1,35,CH) -
HEADER('PRICE OF BOOK') ON(1704,BI,C1,F'$') -
BTITLE('PUBLISHER:') BREAK(106,4,CH) -
BAVERAGE('AVERAGE FOR THIS PUBLISHER') -
BTOTAL('TOTAL FOR THIS PUBLISHER') -
AVERAGE('AVERAGE FOR ALL PUBLISHERS') -
TOTAL('TOTAL FOR ALL PUBLISHERS') -
```

Printing Sectioned Reports



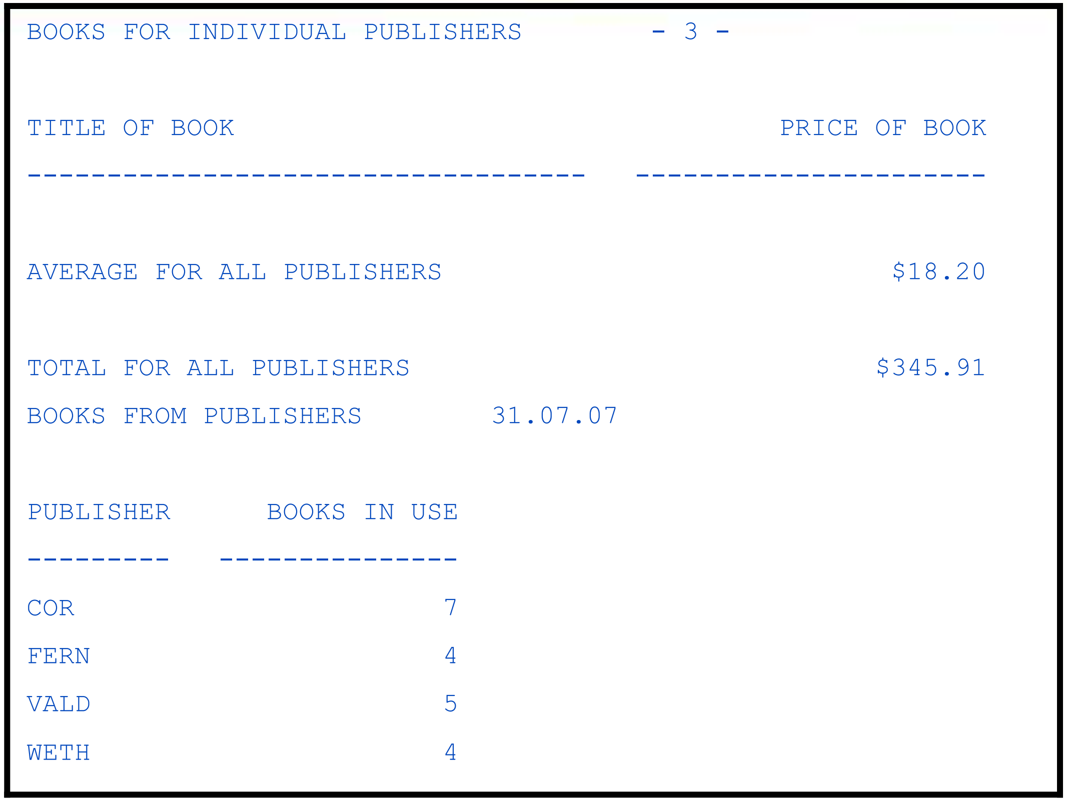
```
BOOKS FOR INDIVIDUAL PUBLISHERS        - 1 -


PUBLISHER:  VALD


TITLE OF BOOK                                      PRICE OF BOOK
-----------------------------------    ----------------------
CELLS AND HOW THEY WORK                                   $24.95
COMPLETE SPANISH DICTIONARY                                $6.50
EDITING SOFTWARE MANUALS                                  $14.50
FREUD'S THEORIES                                          $12.50
INTRODUCTION TO BIOLOGY                                   $23.50
NOVEL IDEAS                                               $24.50
SHORT STORIES AND TALL TALES                              $15.20
STRATEGIC MARKETING                                       $23.50
VIDEO GAME DESIGN                                         $21.99
ZEN BUSINESS                                              $12.00


AVERAGE FOR THIS PUBLISHER                                $17.91


TOTAL FOR THIS PUBLISHER                                 $179.14
```

Printing Sectioned Reports



```
BOOKS FOR INDIVIDUAL PUBLISHERS        - 2 -

PUBLISHER:  WETH

TITLE OF BOOK                                  PRICE OF BOOK
-----------------------------------   ----------------------
ANTICIPATING THE MARKET                               $20.00
CIVILIZATION SINCE ROME FELL                          $13.50
COMPUTERS: AN INTRODUCTION                            $18.99
EIGHTEENTH CENTURY EUROPE                             $17.90
GUIDE TO COLLEGE LIFE                                 $20.00
GUNTHER'S GERMAN DICTIONARY                           $10.88
REBIRTH FROM ITALY                                    $25.60
SYSTEM PROGRAMMING                                    $31.95
THE INDUSTRIAL REVOLUTION                              $7.95

AVERAGE FOR THIS PUBLISHER                            $18.53

TOTAL FOR THIS PUBLISHER                             $166.77
```

Printing Sectioned Reports



```
BOOKS FOR INDIVIDUAL PUBLISHERS        - 3 -


TITLE OF BOOK                                  PRICE OF BOOK

-----------------------------------   ----------------------



AVERAGE FOR ALL PUBLISHERS                            $18.20


TOTAL FOR ALL PUBLISHERS                             $345.91

BOOKS FROM PUBLISHERS        31.07.07


PUBLISHER      BOOKS IN USE

---------   ---------------

COR                       7

FERN                      4

VALD                      5

WETH                      4
```

Additional ICETOOL slides

More fun with ICETOOL

```
//TOOLIN   DD *
* STATISTICS FROM ALL BRANCHES
     STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
* BOOKS FROM VALD AND WETH
     SORT FROM(BKS) TO(DAPUBS,PRPUBS) USING(SPUB)
* SEPARATE OUTPUT FOR CALIFORNIA AND COLORADO BRANCHES
     SORT FROM(ALL) USING(CACO)
* CALIFORNIA BRANCHES PROFIT ANALYSIS
     RANGE FROM(CADASD) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)
* BRANCHES WITH LESS THAN 32 EMPLOYEES
     RANGE FROM(ALL) ON(18,4,ZD) LOWER(32)
* PRINT PROFIT, EMPLOYEES, AND CITY FOR EACH COLORADO BRANCH
     DISPLAY FROM(CODASD) LIST(OUT) ON(28,6,PD) ON(18,4,ZD)  ON(1,15,CH)
* PRINT A REPORT FOR THE COLORADO BRANCHES
   DISPLAY FROM(CODASD) LIST(RPT) -
       DATE TITLE('COLORADO BRANCHES REPORT') PAGE -
       HEADER('CITY') HEADER('PROFIT') HEADER('EMPLOYEES') -
       ON(1,15,CH) ON(28,6,PD) ON(18,4,ZD) BLANK -
       TOTAL('TOTAL') AVERAGE('AVERAGE') MINIMUM('LOWEST')
* PRINT A REPORT OF BOOKS FOR INDIVIDUAL PUBLISHERS
   DISPLAY FROM(DAPUBS) LIST(SECTIONS) -
       TITLE('BOOKS FOR INDIVIDUAL PUBLISHERS') PAGE -
       HEADER('TITLE OF BOOK')  ON(1,35,CH) -
       HEADER('PRICE OF BOOK') ON(170,4,BI,C1,F'$') -
       BTITLE('PUBLISHER:') BREAK(106,4,CH) -
       BAVERAGE('AVERAGE FOR THIS PUBLISHER') -
       BTOTAL('TOTAL FOR THIS PUBLISHER') -
       AVERAGE('AVERAGE FOR ALL PUBLISHERS') -
TOTAL('TOTAL FOR ALL PUBLISHERS')
* PRINT THE COUNT OF BOOKS IN USE FROM EACH PUBLISHER
   OCCUR FROM(BKIN) LIST(PUBCT) BLANK -
      TITLE('BOOKS FROM PUBLISHERS') DATE(DMY.) -
      HEADER('PUBLISHER') HEADER('BOOKS IN USE') -
      ON(106,4,CH) ON(VALCNT)
```

More fun with ICETOOL, continued

```
* SEPARATE OUTPUT CONTAINING RECORDS FOR PUBLISHERS
* WITH MORE THAN 4 BOOKS IN USE
     SELECT FROM(BKIN) TO(BKOUT) ON(106,4,CH) HIGHER(4)
/*
//ALL    DD DSN=S744428.SORT.BRANCH,DISP=SHR
//BKS    DD DSN=S744428.SORT.SAMPIN,DISP=SHR
//       DD DSN=S744428.SORT.SAMPADD,DISP=SHR
//DAPUBS   DD DSN=L2.SAMPLE.SORT.DAPUBS3,
//            DISP=(NEW,CATLG),SPACE=(CYL,(5,5)),UNIT=SYSDA
//PRPUBS   DD DSN=L2.SAMPLE.SORT.PRPUBS3,
//            DISP=(NEW,CATLG),SPACE=(CYL,(5,5)),UNIT=SYSDA
//SPUBCNTL DD *
   SORT FIELDS=(106,4,A,1,75,A),FORMAT=CH
    INCLUDE COND=(106,4,EQ,C'VALD',OR,106,4,EQ,C'WETH'),
            FORMAT=CH
/*
//CACOCNTL DD *
     SORT FIELDS=(1,15,CH,A)
     OUTFIL FNAMES=(CADASD,CATAPE),INCLUDE=(16,2,CH,EQ,C'CA')
     OUTFIL FNAMES=(CODASD,COTAPE),INCLUDE=(16,2,CH,EQ,C'CO')
/*
//CADASD DD  DSN=&&CA,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//CATAPE DD DSN=CA.BRANCH,UNIT=VTAPE,
//  DISP=(NEW,KEEP)
//CODASD DD  DSN=&&CO,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//COTAPE DD DSN=CO.BRANCH,UNIT=VTAPE,
//  DISP=(NEW,KEEP)
//OUT DD SYSOUT=*
//RPT DD SYSOUT=*
//SECTIONS DD SYSOUT=*
//BKIN DD DSN=S744428.SORT.SAMPIN,DISP=SHR
//PUBCT DD SYSOUT=*
//BKOUT DD DSN=S744428.BOOKS1,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(3,3)),UNIT=3390
/*
```

Some practical uses of DFSORT for L2

- The following JOB was used to SORT MCD records in a MCDS with a specific VOLSER

```
//SORTIN   DD  DSN=L2.SAMPLE.HSMDATA.MCDS,DISP=SHR
//SORTOUT  DD DSN=L2.SAMPLE.HSMDATA2.OUTPUT,
//            DISP=(NEW,CATLG),SPACE=(CYL,(5,5)),UNIT=SYSDA
//MSGOUT1  DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SYSIN    DD  *
    OPTION VLSHRT
    MERGE FIELDS=COPY
    RECORD TYPE=V
    INCLUDE COND=((51,1,BI,EQ,X'00'),AND,
           (69,6,CH,EQ,C'A00109'))
/*
```

•JOBLOG can be located on SNJMAS3

→**L2.SAMPLE.SORT.HSMDATA.JOBLOG**

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7-
***********************************************************************
HSMATH0.A.GDG.G0003V00                      .M..{ÜLð6pb¯{8Ü"çr§£A00109d
HSMATH0.A.GDG.G0005V00                      .M..{ÜLý..ÝÛ{8ÚÞ..ÙøA00109d
HSMATH0.B.GDG.G0004V00                      .M..{ÜLq3J¥ø{Ü}ÿYìÌ¾A00109d
HSMATH0.B.GDG.G0005V00                      .M..{ÜLÆì>.¯{Ü}0.Û92A00109d
HSMATH0.B.GDG.G0006V00                      .M..{ÜLv.<.ø{ÜJ.PôpÊA00109d
HSMATH0.DS.MIGRATED.A                       .M..{ÜLý.)..{5;Ú.Ñ£©A00109d
HSMATH0.MEDIUM.MAKE05C                      .M..{ÜOy}...{ÜNcTj.ÚA00109d
HSMATH0.MEDIUM.MAKE05I                      .M..{ÜOC¥s.Î{ÜNB%.¶.A00109d
***********************************************************************
```

Some practical uses of DFSORT for L2

- This JOB was created to SORT TTOCs from an OCDS that point to a specific VOLSER

```
//SORTOUT   DD DSN=L2.SAMPLE.HSMDATA2.OUTPUT,
//             DISP=(NEW,CATLG),SPACE=(CYL,(5,5)),UNIT=SYSDA
//MSGOUT1   DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//SYSIN     DD  *
    OPTION VLSHRT
    MERGE FIELDS=COPY
    RECORD TYPE=V
    INCLUDE COND=(5,1,BI,EQ,X'33')
    OUTFIL FNAMES=SORTOUT,VLFILL=X'40',
     OUTREC=(1,4,C'    HSEND FIXCDS E ',6,8,C' DELETE')
    END
/*
```

```
----+----1----+----2----+----3----+----4
****************************************
    HSEND FIXCDS E M-A00016 DELETE
    HSEND FIXCDS E M-A00109 DELETE
    HSEND FIXCDS E M-A00127 DELETE
    HSEND FIXCDS E M-A00145 DELETE
****************************************
```

- JOBLOG can be located on SNJMAS3 →**L2.SAMPLE.SORT.HSMDATA2.JOBLOG**

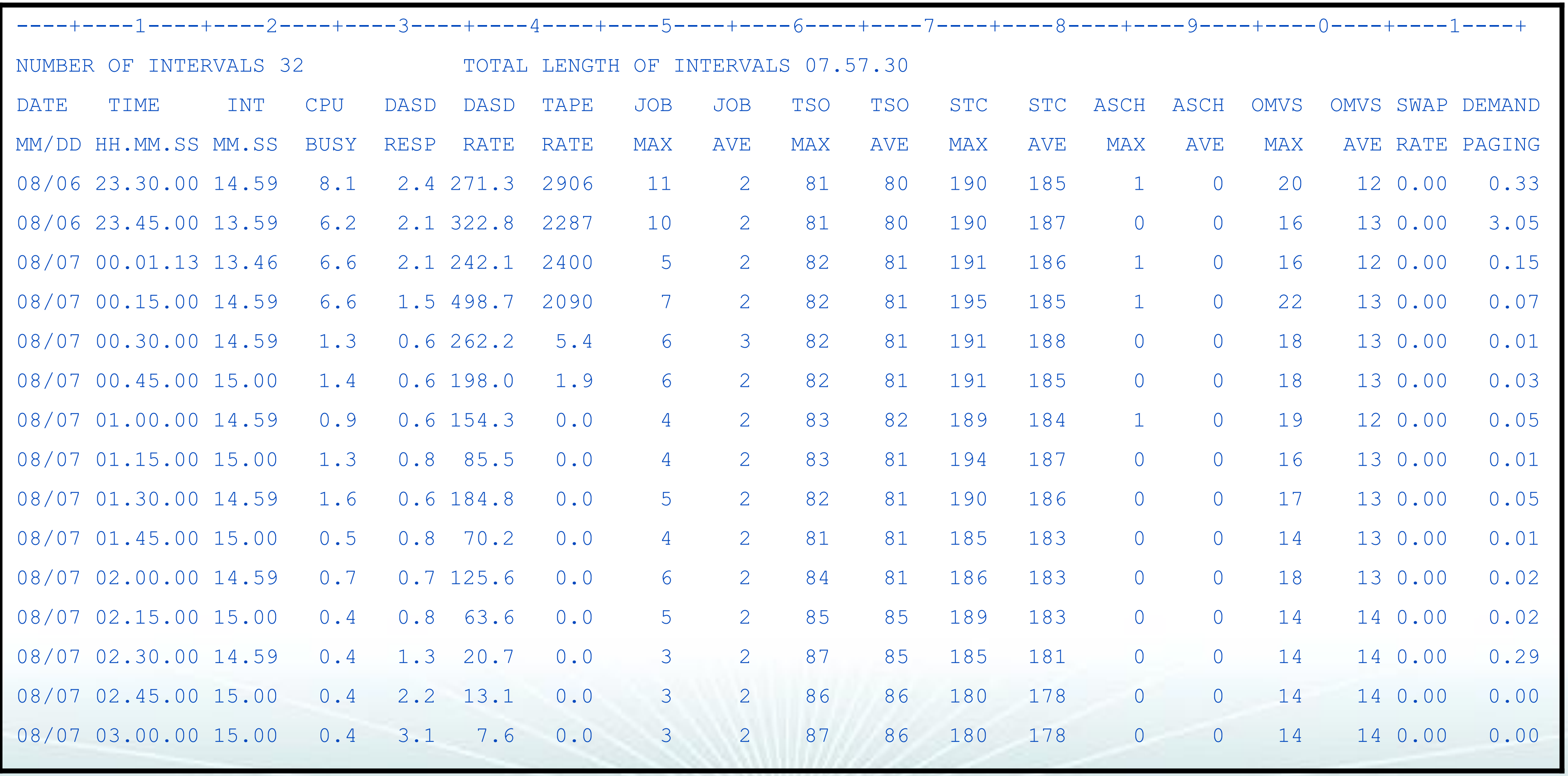
Some practical uses of DFSORT for L2

- The following JOB basically extracts RMF records from an SMF file and puts them in a date-time sequence

```
//SORT  EXEC PGM=SORT,REGION=0M
//SYSPRINT DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//SORTIN   DD DISP=SHR,DSN=L2.SAMPLE.SORT.DATA.SMFDATA
//SORTOUT  DD  DISP=(NEW,PASS),DSN=&&SORTRMF,UNIT=SYSDA,
//             SPACE=(CYL,(200,200),RLSE),DCB=*.SORTIN
//SYSIN    DD   *
*  SORT ON SMF DATE AND TIME
   SORT FIELDS=(11,4,PD,A,7,4,BI,A)
*  INCLUDE ONLY RMF TYPES 70-78
   INCLUDE COND=(6,1,BI,GE,X'46',AND,6,1,BI,LE,X'4E')
   OPTION VLSHRT,DYNALLOC=(SYSDA,8)
/*
//RMFSUM   EXEC PGM=ERBRMFPP,REGION=0M
//MFPMSGDS DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//MFPINPUT DD DSN=&&SORTRMF,DISP=SHR
//SYSIN DD *
    SYSOUT(H)
    SUMMARY(TOT,INT)
    STOD(0000,2400)
/*
```

Some practical uses of DFSORT for L2, output from SMF/RMF data

- JOBLOG can be located on SNJMAS3 →**L2.SAMPLE.SORT.SMFDATA.JOBLOG**



```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0----+----1----+
NUMBER OF INTERVALS 32            TOTAL LENGTH OF INTERVALS 07.57.30
DATE   TIME     INT   CPU   DASD  DASD  TAPE   JOB   JOB   TSO   TSO   STC   STC  ASCH  ASCH  OMVS  OMVS SWAP DEMAND
MM/DD HH.MM.SS MM.SS  BUSY  RESP  RATE  RATE   MAX   AVE   MAX   AVE   MAX   AVE   MAX   AVE   MAX   AVE RATE PAGING
08/06 23.30.00 14.59   8.1   2.4 271.3  2906    11     2    81    80   190   185     1     0    20    12 0.00   0.33
08/06 23.45.00 13.59   6.2   2.1 322.8  2287    10     2    81    80   190   187     0     0    16    13 0.00   3.05
08/07 00.01.13 13.46   6.6   2.1 242.1  2400     5     2    82    81   191   186     1     0    16    12 0.00   0.15
08/07 00.15.00 14.59   6.6   1.5 498.7  2090     7     2    82    81   195   185     1     0    22    13 0.00   0.07
08/07 00.30.00 14.59   1.3   0.6 262.2   5.4     6     3    82    81   191   188     0     0    18    13 0.00   0.01
08/07 00.45.00 15.00   1.4   0.6 198.0   1.9     6     2    82    81   191   185     0     0    18    13 0.00   0.03
08/07 01.00.00 14.59   0.9   0.6 154.3   0.0     4     2    83    82   189   184     1     0    19    12 0.00   0.05
08/07 01.15.00 15.00   1.3   0.8  85.5   0.0     4     2    83    81   194   187     0     0    16    13 0.00   0.01
08/07 01.30.00 14.59   1.6   0.6 184.8   0.0     5     2    82    81   190   186     0     0    17    13 0.00   0.05
08/07 01.45.00 15.00   0.5   0.8  70.2   0.0     4     2    81    81   185   183     0     0    14    13 0.00   0.01
08/07 02.00.00 14.59   0.7   0.7 125.6   0.0     6     2    84    81   186   183     0     0    18    13 0.00   0.02
08/07 02.15.00 15.00   0.4   0.8  63.6   0.0     5     2    85    85   189   183     0     0    14    14 0.00   0.02
08/07 02.30.00 14.59   0.4   1.3  20.7   0.0     3     2    87    85   185   181     0     0    14    14 0.00   0.29
08/07 02.45.00 15.00   0.4   2.2  13.1   0.0     3     2    86    86   180   178     0     0    14    14 0.00   0.00
08/07 03.00.00 15.00   0.4   3.1   7.6   0.0     3     2    87    86   180   178     0     0    14    14 0.00   0.00
```

Calling DFSORT from a COBOL program

```
//EXAMP    JOB  A492,PROGRAMMER
//BOOKS    EXEC PGM=COBOLPGM
//STEPLIB  DD   DSN=A492.SM,DISP=SHR
//         DD   DSN=USER.PGMLIB,DISP=SHR
//SYSOUT   DD   SYSOUT=A
//MASTIN   DD   DSN=A123456.MASTER,DISP=OLD
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//MASTOUT  DD   DSN=A123456.OUTB,DISP=(NEW,CATLG,DELETE),
//         SPACE=(CYL,(1,1)),UNIT=SYSDA
//PRINTFL  DD   SYSOUT=A
/*
```

- The program name on the **EXEC** statement is that of the **COBOL** program.
- The **STEPLIB DD** statement defines the **library** containing the DFSORT program, as well as the library containing the COBOL program.
- The name of the DD statement for the input file need not be **SORTIN**.
- The name of the DD statement for the output file need not be **SORTOUT**.

Calling DFSORT from a PL/1 program

```
//EXAMP    JOB  A492,PROGRAMMER
//BOOKS    EXEC PGM=PLIPGM
//STEPLIB  DD   DSN=A492.SM,DISP=SHR
//         DD   DSN=USER.PGMLIB,DISP=SHR
//SYSOUT   DD   SYSOUT=A
//SORTIN   DD   DSN=A123456.SORT.SAMPIN,DISP=SHR
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTOUT  DD   DSN=A123456.SORT.SAMPOUT,DISP=OLD
//          SPACE=(CYL,(1,1)),UNIT=SYSDA
//SORTCNTL DD *
    INCLUDE COND=(110,5,CH,EQ,C'ENGL')
//SYSPRINT DD   SYSOUT=A
/*
```

- When calling DFSORT, a **PL/I** program must pass a **SORT control statement**
- A **RECORD control statement**
- The amount of **main storage** to be allocated to DFSORT
- You can also pass control statements by using the **SORTCNTL DD** statement

Listing your systems installation defaults (A.K.A. – ICEMAC)

- To generate a list of your installation defaults run the following JOB

```
//ICEMAC    EXEC PGM=ICETOOL
//TOOLMSG   DD   SYSOUT=*
//DFSMSG    DD   SYSOUT=*
//LIST1     DD   SYSOUT=*
//TOOLIN    DD   *
 DEFAULTS LIST(LIST1)
/*
```

- This will get you a complete listing of DFSORT installation defaults for all 4 environments
- JOBLOG can be located on SNJMAS3 →**L2.SAMPLE.SORT.ICEMAC.JOBLOG**

Output from ICEMAC JOB

```
ITEM           JCL(ICEAM1)             INV(ICEAM2)             TSO(ICEAM3)             TSOINV(ICEAM4)
----------     --------------------    --------------------    --------------------    --------------
RELEASE        V1R5                    V1R5                    V1R5                    V1R5
MODULE         ICEAM1                  ICEAM2                  ICEAM3                  ICEAM4
APAR LEVEL     BASE                    BASE                    BASE                    BASE
COMPILED       07/21/03                07/21/03                07/21/03                07/21/03

ENABLE         NONE                    NONE                    NONE                    NONE

ABCODE         MSG                     MSG                     MSG                     MSG
ALTSEQ         SEE BELOW               SEE BELOW               SEE BELOW               SEE BELOW
ARESALL        0                       0                       0                       0
ARESINV        NOT APPLICABLE          0                       NOT APPLICABLE          0
CFW            YES                     YES                     YES                     YES
CHALT          NO                      NO                      NO                      NO
CHECK          YES                     YES                     YES                     YES
CINV           YES                     YES                     YES                     YES
COBEXIT        COB2                    COB2                    COB2                    COB2
DIAGSIM        NO                      NO                      NO                      NO
DSA            64                      64                      64                      64
DSPSIZE        MAX                     MAX                     MAX                     MAX
DYNALOC         (SYSDA,4)               (SYSDA,4)               (SYSDA,4)               (SYSDA,4)
```

Run-time overrides

- You can change a number of DFSORT parameters at run-time using a DFSPAM DD statement
- Please NOTE →You cannot override all DFSORT parameters but you can change many of them
- Reference the DFSORT Application Programming Guide to determine what parameters can be changed at run-time

```
//DFSPARM DD *
 OPTION MOSIZE=0, MAINSIZE=10M
```

- The preceding run-time overrides tell DFSORT to turn off Memory Object Sorting (MOSIZE=0) and to set the available main storage to 10M for this particular JOB.

References

- *z/OS V1R6.0 DFSORT Application Programming Guide*
  - ***Document Number:*** *SC26-7523-01*
  - http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ICE1CA10/CCONTENTS?DT=20050222160456
- *Getting Started with DFSORT*
  - ***Document Number:*** *SC26-410907*
  - http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/ICECG202/CCONTENTS?DT=19950424102033
- **DFSORT:ICETOOL Mini-User Guide**
  - http://www-304.ibm.com/jct01004c/systems/support/storage/software/sort/mvs/icetool/pdf/sorttool.pdf

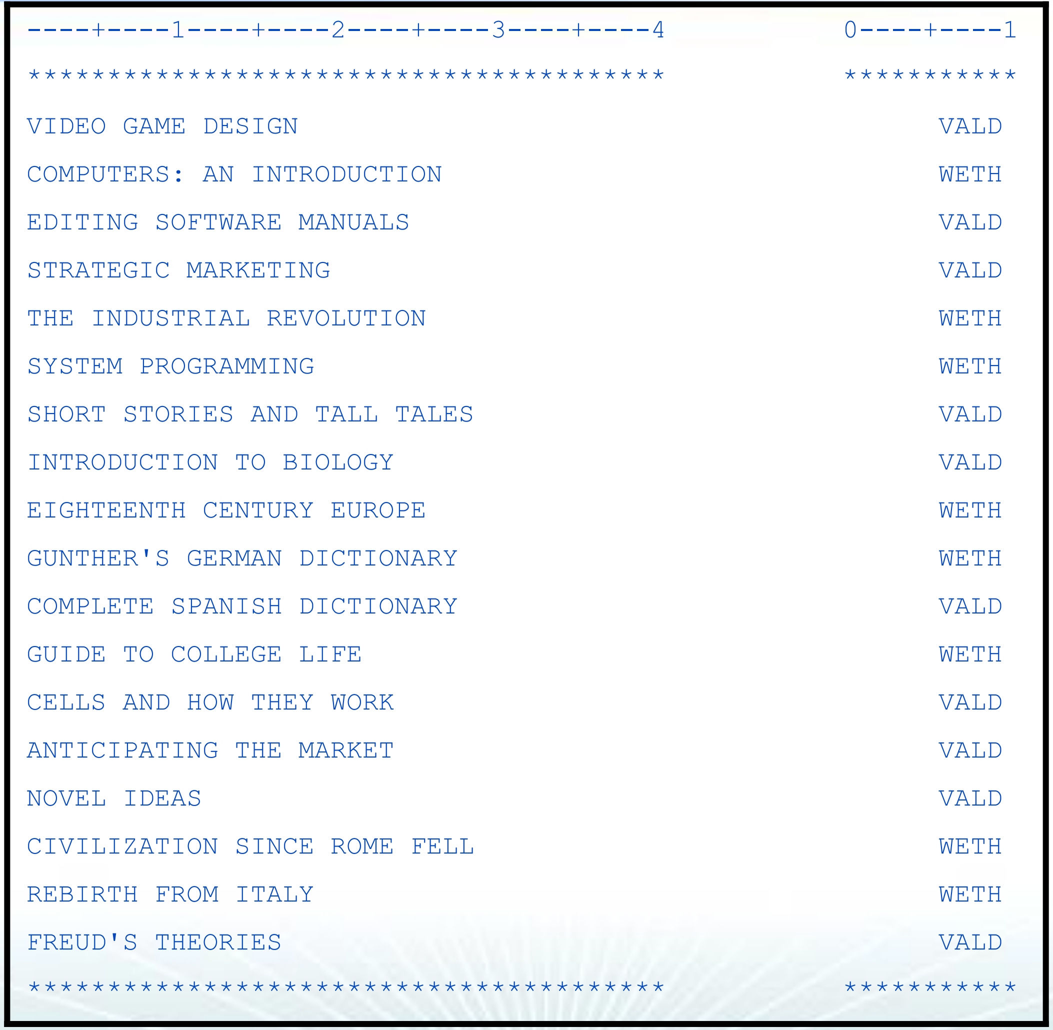
References

- **L2.SAMPLE.SORT.EXAMPLE.JOBS**
  - PDS dataset that contains all of the JOBs that were used in this presentation

- **L2.SAMPLE.SORT.DATA***
  - Input datasets used for examples in this presentation

- **L2.SAMPLE***
  - JOBLOGs and datasets created by sample JOBs

Data prior to running ICETOOL, from datasets SAMPIN and SAMPADD



```
----+----1----+----2----+----3----+----4           0----+----1
****************************************           ***********
VIDEO GAME DESIGN                                        VALD
COMPUTERS: AN INTRODUCTION                               WETH
EDITING SOFTWARE MANUALS                                 VALD
STRATEGIC MARKETING                                      VALD
THE INDUSTRIAL REVOLUTION                                WETH
SYSTEM PROGRAMMING                                       WETH
SHORT STORIES AND TALL TALES                             VALD
INTRODUCTION TO BIOLOGY                                  VALD
EIGHTEENTH CENTURY EUROPE                                WETH
GUNTHER'S GERMAN DICTIONARY                              WETH
COMPLETE SPANISH DICTIONARY                              VALD
GUIDE TO COLLEGE LIFE                                    WETH
CELLS AND HOW THEY WORK                                  VALD
ANTICIPATING THE MARKET                                  VALD
NOVEL IDEAS                                              VALD
CIVILIZATION SINCE ROME FELL                             WETH
REBIRTH FROM ITALY                                       WETH
FREUD'S THEORIES                                         VALD
****************************************           ***********
```