

#SHAREorg



# DFSMS DFSORT: The ICETOOL Cometh - Getting Started Using ICETOOL

Sri Hari Kolusu  
IBM

Wednesday, February 6<sup>th</sup> 2013  
Session Number : 12983

Skolusu@us.ibm.com



# Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.



<b>AI X®</b>	<b>DB2®</b>	<b>DFSORT</b>	<b>IBM®</b>	<b>Language Environment®</b>	<b>Redbooks®</b>	<b>System Storage</b>	<b>System z10 Business Class</b>	<b>z10 EC</b>
<b>Bl eadCenter®</b>	<b>DFSMS</b>	<b>Damino®</b>	<b>IBM eServer</b>	<b>MVS</b>	<b>REXX</b>	<b>System x®</b>	<b>Tivoli®</b>	<b>zEnterprise®</b>
<b>DocManager®</b>	<b>DFSMSdss</b>	<b>DSS000</b>	<b>IBM Logo®</b>	<b>Parallel Sysplex®</b>	<b>RMF</b>	<b>System z</b>	<b>WebSphere®</b>	<b>zSeries®</b>
<b>QIC®</b>	<b>DFSMSshm</b>	<b>DSS000®</b>	<b>IMS</b>	<b>ProductPac®</b>	<b>ServerPac®</b>	<b>System z9</b>	<b>z10</b>	
<b>DataPower®</b>	<b>DFSMSrsm</b>	<b>FIDC®</b>	<b>InfiniBand</b>	<b>RMCF®</b>	<b>SYSREXX</b>	<b>System z10</b>	<b>z10 EC</b>	

\* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.  
 IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.  
 Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.  
 Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.  
 Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.  
 Windows Server and the Windows logo are trademarks of the Microsoft group of countries.  
 ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.  
 UNIX is a registered trademark of The Open Group in the United States and other countries.  
 Java and all Java based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.  
 Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.  
 Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.  
 \* Other product and service names might be trademarks of IBM or other companies.

**Notes:**

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.  
 IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.  
 All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.  
 This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.  
 All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.  
 Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.  
 Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.  
 This information provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g. zIIPs, zAAPs, and IFLs) ("SEs"). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at [www.ibm.com/systems/support/machine\\_warranties/machine\\_code/aut.html](http://www.ibm.com/systems/support/machine_warranties/machine_code/aut.html) ("AUT"). No other workload processing is authorized for execution on an SE. IBM offers SE at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

2 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



# System z Social Media Channels



- **Top Facebook pages related to System z:**
  - [IBM System z](#)
  - [IBM Academic Initiative System z](#)
  - [IBM Master the Mainframe Contest](#)
  - [IBM Destination z](#)
  - [Millennial Mainframer](#)
  - [IBM Smarter Computing](#)
- **Top LinkedIn groups related to System z:**
  - [System z Advocates](#)
  - [SAP on System z](#)
  - [IBM Mainframe- Unofficial Group](#)
  - [IBM System z Events](#)
  - [Mainframe Experts Network](#)
  - [System z Linux](#)
  - [Enterprise Systems](#)
  - [Mainframe Security Gurus](#)
- **Twitter profiles related to System z:**
  - [IBM System z](#)
  - [IBM System z Events](#)
  - [IBM DB2 on System z](#)
  - [Millennial Mainframer](#)
  - [Destination z](#)
  - [IBM Smarter Computing](#)
- **YouTube accounts related to System z:**
  - [IBM System z](#)
  - [Destination z](#)
  - [IBM Smarter Computing](#)

- **Top System z blogs to check out:**
  - [Mainframe Insights](#)
  - [Smarter Computing](#)
  - [Millennial Mainframer](#)
  - [Mainframe & Hybrid Computing](#)
  - [The Mainframe Blog](#)
  - [Mainframe Watch Belgium](#)
  - [Mainframe Update](#)
  - [Enterprise Systems Media Blog](#)
  - [Dancing Dinosaur](#)
  - [DB2 for z/OS](#)
  - [IBM Destination z](#)
  - [DB2utor](#)



3 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



## Agenda

- What is ICETOOL?
- Generating Sample data
- Basic ICETOOL JCL
- ICETOOL Utility Operators
- ICETOOL Operator Syntax
- Cool Things you can do with ICETOOL
- Defining and Using Symbols
- Q & A

The purpose of this session is to provide a brief overview, usage, invocation and examples of various cool tasks you can perform using IBM's Flagship product DFSORT's ICETOOL. You will learn about ICETOOL's JCL and control statements while writing a large "main" ICETOOL job that uses many of the ICETOOL operators, as well as several additional smaller ICETOOL jobs that illustrate specific points.

## What is ICETOOL?

- ICETOOL is a batch front-end utility
- ICETOOL uses the capabilities of DFSORT
- ICETOOL can be called directly or from a program
- ICETOOL includes 17 operators
- Messages and return codes



ICETOOL is a versatile data set processing and reporting utility that provides an easy-to-use batch front-end for DFSORT. ICETOOL combines new features with previously available DFSORT features to perform complex sorting, copying, merging, reporting and analytical tasks using multiple data sets in a single job step. 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. This presentation introduces you to ICETOOL's 17 "operators", which allow you to do a wide variety of tasks.

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 sets a return code for each operation it performs. The return codes are:

- 0 - Successful completion. No errors were detected.
- 4 - Successful completion. DFSORT detected one or more warning conditions, or RC4 was used for COUNT and the criteria was met.
- 8 - Unsuccessful completion. RC8 was used for COUNT and the criteria was met.
- 12 - Unsuccessful completion. ICETOOL detected one or more errors, or RC12 was used (or defaulted) for COUNT and the criteria was met.
- 16 - Unsuccessful completion. DFSORT detected one or more errors.
- 20 - Message data set error. The TOOLMSG DD statement was not present or the TOOLMSG data set was not opened.
- 24 - Unsupported operating system. This operating system is not supported by this release of DFSORT.

You will learn about ICETOOL's JCL and control statements in the next few slides.

## Generating Sample data

- Before we get started, this presentation uses a chain of bookstores for examples
  - Colorado and California
- Corresponding info in Getting Started
- **ICEDATA JOB**
  - Creates all data used in examples
  - Located in **SYS1.SICESAMP**

z/OS DFSORT: Getting Started manual can be found at  
<http://www.ibm.com/support/docview.wss?rs=114&uid=isg3T7000080>

## Basic ICETOOL JCL

- ICETOOL required JCL statements
  - TOOLMSG DD
  - DFSMSG or SSMSG DD
  - TOOLIN DD
  - XXXXCNTL DD
  - ICETOOL statements indicate operations to be performed

The JCL statements that are required as a result of the specified operator statements.

The TOOLMSG statement defines the output data set for ICETOOL messages.

The DFSMSG or SSMSG statement defines the output data set for DFSORT messages.

TOOLIN is where you create ICETOOL statements using operators.

XXXXCNTL is where you place sort control statements, XXXX can be anything, but CNTL is required.

## Basic ICETOOL JCL (continued)



```
//EXAMP      JOB A492,PROGRAMMER
//STEP0100 EXEC PGM=ICETOOL
//TOOLMSG    DD SYSOUT=*
//DFSMSG     DD SYSOUT=*
//TOOLIN     DD *
<ICETOOL statements go here>
...
<Additional JCL statements go here, DD's, etc...>
...
//XXXXCNTL DD *
<Sort control statements go here>
...
```

8 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



EXEC PGM=ICETOOL - to use ICETOOL directly.

TOOLMSG - ICETOOL message data set. This data set has the same attributes as the DFSORT SYSOUT data set.

DFSMSG - DFSORT message data set. This data set has the same attributes as the DFSORT SYSOUT data set.

TOOLIN - ICETOOL statements. This data set has the same attributes as the DFSORT SYSIN data set.

Additional JCL - as required for operators you specify



## ICETOOL Utility operators



- COPY
- COUNT
- DATASORT
- DEFAULTS
- DISPLAY
- MERGE
- MODE
- OCCUR
- RANGE

9 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



The 17 ICETOOL operators listed below can be used to perform a variety of functions.

**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. COUNT can also be used to create an output data set containing text and the count, or to set RC=12, RC=8, RC=4, or RC=0 based on meeting criteria for the number of records in a data set (for example, empty, not empty, less than, equal to, or greater than 5000 records, and so on).

**DATASORT** Sorts data records between header and trailer records in a data set to an output data set.

**DEFAULTS** Prints the DFSORT installation defaults in a separate list data set.

**DISPLAY** Prints the values and characters of specific numeric and character fields in a separate list data set. Simple, tailored or sectioned reports can be produced.

**MERGE** Merges one or more data sets to one or more output data sets.

**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.

## ICETOOL Utility operators (continued)



- **RESIZE**
- **SELECT**
- **SORT**
- **SPLICE**
- **STATS**
- **SUBSET**
- **UNIQUE**
- **VERIFY**

10 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



**RESIZE** Creates a larger record from multiple shorter records, or creates multiple shorter records from a larger record, that is, resizes fixed length records.

**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). Records that are not selected can be saved in a separate output data set.

**SORT** Sorts a data set to one or more output data sets.

**SPLICE** Splices together specified fields from records that have the same specified numeric or character field values (that is, duplicate values), but different information. Specified fields from two or more records can be combined to create an output record. The fields to be spliced can originate from records in different data sets, so you can use SPLICE to do various "join" and "match" operations.

**STATS** Prints messages containing the minimum, maximum, average, and total for specified numeric fields in a data set.

**SUBSET** Selects records from a data set based on keeping or removing header records, relative records or trailer records. Records that are not selected can be saved in a separate output 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.

## ICETOOL Operator Statement Syntax

- **operator operand ... operand**
- Example: `COPY FROM(IN) TO(OUT1,OUT2)`
  - Operator is one of the seventeen ICETOOL operator names.
  - Operand is keyword or keyword(parameter,...)
  - One or more blanks can be used before the operator and between operands.
  - Columns 1-72 are scanned; columns 73-80 are ignored.
  - Continuation can be indicated by a dash (-) after the operator or any operand. Each operand must be completely specified on one line.
- Example: 

```
SORT FROM(IN1)      -  
          TO(OUT1,OUT2,OUT3) -  
          USING(ABCD)
```

Each ICETOOL operator statement describes a task you want ICETOOL to perform. Any number of operators can be specified and in any order.

## Cool Things you can do with ICETOOL



- **Creating multiple identical copies**
- **Collecting statistics using ICETOOL**
- **The STATS Operator/statistical output**
- **Counting values in a range**
- **Creating Tailored Reports**
- **Edit Masks, Leading Zeros, Edit Patterns and Division**
- **Leading, Floating and Trailing Characters**
- **Printing Sectioned Reports**
- **How Many Times Fields Occur**
- **Records by Field Occurrences**
- **Create small records from large records and vice versa**

12. [complete your sessions evaluation online at SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



The above listed is just a partial list of the cool tricks you can do with ICETOOL. There are several ways to exploit this powerful tool called ICETOOL. By using various combinations of the 17 ICETOOL operators, you can easily create applications that perform many complex tasks.

## Creating multiple identical copies using ICETOOL



```
//STEP0100 EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//TOOLIN DD *
* BOOKS FROM VALD AND WETH
  SORT FROM (BKS) - ←BKS is DD of input
    TO (DAPUBS, PRPUBS) - ←DAPUBS/PRPUBS is DD of first/second output
    USING (SPUB) ←First four characters of control data set DD


/*
//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),RLSE),UNIT=SYSDA
//PRPUBS DD DSN=L2.SAMPLE.SORT.PRPUBS,
// DISP=(NEW,CATLG),SPACE=(CYL,(5,5),RLSE),UNIT=SYSDA
//SPUBCNTL DD *
  INCLUDE COND=(106,4,EQ,C'VALD',OR,106,4,EQ,C'WETH'),FORMAT=CH
  SORT FIELDS=(106,4,A,1,75,A),FORMAT=CH
/*
```

13. Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



Essentially we are using ICETOOL to tell DFSORT to SORT from the DD (BKS) to the DD's (DAPUBS and PRPUBS) using control DD SPUB(our control statements). Please note that the first 4 characters of the control DD can be anything, the last 4 'CNTL' are required. You reference the control DD in the USING operand using whatever 4 characters you provide, in this example 'SPUB'.



## Input data : S744428.SORT.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
*****

```

14. Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



**SHARE**  
in San Francisco  
2013

Sample Input data



## Output Data 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
*****

```

**\*\* Only Portion of the data is shown to emphasize how the data is sorted**



This is how the output looks like based on what was specified in SPUBCNTL. We are including records with character data equal to VALD or WETH at column 106. We are Sorting the records in Ascending order on 2 fields and we are writing out these records in ascending order.

## Collecting Statistics using ICETOOL

```
//TOOLIN DD *
* BOOKS FROM VALD AND WETH
  SORT FROM(BKS) TO(DAPUBS,PRPUBS) USING(SPUB)
* STATISTICS FROM ALL BRANCHES
  STATS FROM(ALL) - ← DD name of the input dataset
                   ON(18,4,ZD) - ← Employees
                   ON(28,6,PD) - ← Profit
                   ON(22,6,PD) ← Revenue
/*
//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),RLSE),UNIT=SYSDA
//PRPUBS DD DSN=L2.SAMPLE.SORT.PRPUBS2,
// DISP=(NEW,CATLG),SPACE=(CYL,(5,5),RLSE),UNIT=SYSDA
//SPUBCNTL DD *
  INCLUDE COND=(106,4,EQ,C'VALD',OR,106,4,EQ,C'WETH'),FORMAT=CH
  SORT FIELDS=(106,4,A,1,75,A),FORMAT=CH
/*
```

- Added 6 lines in **RED**



## Sample Input Data for STATS Operator

```

-----+-----1-----+-----2-----+-----3-----+-----4-----+-----5
***** Top of Data *****
Los Angeles      CA003B....ë.....ý
San Francisco    CA003E...âb.....c.
Fort Collins     CO002B.....f.
Sacramento      CA002I...âE%....%
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 *****

```

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

Sample Input data for the stats operator.

## The Output from STATS Operator



```
* 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 : +0000000000050665
STATISTICS FOR (22,6,PD) :
  MINIMUM: +000000000012300, MAXIMUM: +000000000042820
  AVERAGE: +000000000027469, TOTAL : +000000000329637
OPERATION RETURN CODE: 00
```


As you can see the STATS operator provides a MINIMUM, MAXIMUM, AVERAGE, and TOTAL for each field of the ON operand.

## Counting Values in a Range using ICETOOL



```
//TOOLIN DD *
* SEPARATE OUTPUT FOR CALIFORNIA
  SORT FROM(ALL) USING (CACO)
* STATISTICS FROM ALL BRANCHES
  STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
* CALIFORNIA BRANCHES PROFIT ANALYSIS
  RANGE FROM(CADASD) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)
/*
//ALL DD DSN=S744428.SORT.BRANCH,DISP=SHR
//CADASD DD DSN=&&CA,DISP=(,PASS),SPACE=(CYL,(2,2),RLSE)
//CACOCNTL DD *
  SORT FIELDS=(1,15,CH,A)
  OUTFIL FAMES=CADASD,INCLUDE=(16,2,CH,EQ,C'CA')
/*
```

Now we are taking a look at the RANGE operator in an effort to get a little more granular. We are going to use the California branches to find out how many branches have profit between -1500 and 8000.



## 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: 0000000000000007
ICE631I 0 NUMBER OF VALUES IN RANGE FOR (28,6,PD) : 0000000000000003
ICE602I 0 OPERATION RETURN CODE: 00
```

20 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



SHARE  
in San Francisco  
2013

The output from the RANGE operator is presented in the TOOLMSG dataset. From the above output we see that there is a total of 7 California branches and of those branches, 3 are within the RANGE we specified.

## Creating 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 BETWEEN(5) -
    TOTAL('TOTAL') AVERAGE('AVERAGE') MINIMUM('LOWEST')

/*
//ALL DD DSN=S744428.SORT.BRANCH,DISP=SHR
/*
//CACOCNTL DD *
  SORT FIELDS=(1,15,CH,A)
  OUTFIL FNames=CODASD,INCLUDE=(16,2,CH,EQ,C'CO')
/*
//CODASD DD DSN=&&CO,DISP=(,PASS),SPACE=(CYL,(2,2)),UNIT=3390
//RPT DD SYSOUT=*
```

21 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



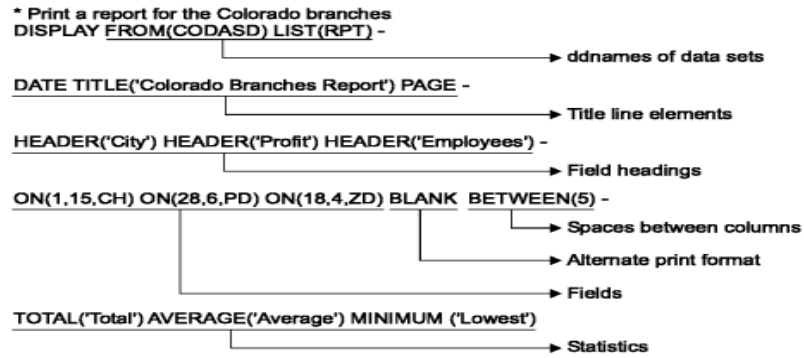
Now we are going to add some bells and whistles using the DISPLAY Operator.

We are going to add a DATE, TITLE, and page number.

We are then going to add some HEADERS so our report looks nice.

We are going to collect some statistics and perform some calculations.

# Creating Tailored Reports



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

This is a breakdown of the TOOLIN statements.



## Tailored Report Output

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-nine pre-defined Edit Masks
  - **d** - decimal digit (0-9)
  - **w** - leading sign blank for + or – for negative
  - **x** - trailing sign blank for + or – for negative
  - **y** - leading sign blank for + or ( for negative
  - **Z** - trailing sign blank for + or ) for negative
- Edit Pattern Mask E1 would look like
  - yd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,dddz
  - 12345678
  - 12,345,678 ←results

24 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



**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

The various edit mask patterns are shown in Table 74 in DFSORT Application Programming Guide.



## Edit Masks Example

Add this Edit Pattern Mask

Changing ON(28,6,PD) to ON(28,6,PD,E1)

CITY	PROFIT	EMPLOYEES
Aspen	5,200	20
Boulder	7,351	32
Denver	6,288	33
<b>Fort Collins</b>	<b>(2,863)</b>	<b>22</b>
Vail	5,027	19
TOTAL	21,003	126
AVERAGE	4,200	25
<b>LOWEST</b>	<b>(2,863)</b>	<b>19</b>

So by adding the E1 EDIT MASK the negative values are now encapsulated in ( ) and commas have been added to make the numbers more readable.

## 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 (without LZ)	Leading zeros (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

This example shows the use of the LZ Edit Mask.

## Edit Patterns

- The pattern (1 to 44 characters) must be enclosed in single apostrophes. Each **9** in the pattern (up to 31) is replaced by a corresponding digit from the numeric value. Characters other than 9 in the pattern appear as specified. To include a single apostrophe (') in the pattern, specify two single apostrophes (").
  - For Example:
    - 8-byte ZD date in the form *mmdyyy* cols. 41-48
    - *mm/dd/yyyy* using ON(41,8,ZD,E'99/99/9999')
    - 01122013 is displayed as **01/12/2013**
  - Or:
    - 10-byte ZD phone number in the form *aaappnnnn* cols. 21-30
    - (aaa)-ppp-nnnn using ON(21,10,ZD, E'(999)-999-9999')
    - 0123456789 is displayed as **(012)-345-6789**

There may be times when Edit MASKS are not particularly useful for unsigned numeric data such as telephone numbers, Social Security numbers, dates, time-of-day, etc. We have Edit patterns for our rescue. For example, 0123456789 is shown as (012)-345-6789 with ON(21,10,ZD,E'(999)-999-9999'). If you have an 8-byte ZD date in the form *mmdyyy* in positions 41-48, you can display it as *mm/dd/yyyy* using ON(41,8,ZD,E'99/99/9999').

An 8-byte value of 03122004 is displayed as **03/12/2004**.

## 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)

For Example:

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

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

For example, you may want the reports have totals rounded to the nearest million or total byte count in Mega/Tera bytes rather than showing the values in bytes. DISPLAY operator has a Division built into it. It specifies division of the numeric data for this field before formatting. x indicates the division factor to be used. The resulting values are rounded down to the nearest integer. Statistics (TOTAL, MAXIMUM, MINIMUM, AVERAGE, BTOTAL, BMAXIMUM, BMINIMUM, BAVERAGE) and column widths reflect the divided numbers For the example in this slide, we collected statistics on the profit field and then divided that by 1000000 which shows the profit or loss in millions.

## Leading, Floating and Trailing Characters

- Add floating, leading, and trailing characters to your numeric and character fields as follows:

- **F'string'** - a floating string
  - Left of the first non-blank character
- **L'string'** - a leading string
  - Beginning of the character or numeric data
- **T'string'** - a trailing string
  - End of the character or numeric data

For Example :

```

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

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

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

In this case we are adding a \$ on the left and trailing \*\* on the right.

## Printing Sectioned Reports



- BREAK operand of DISPLAY
  - Create reports divided into sections
  - BREAK on a character or numeric
  - Previously sorted data
- **BREAK(p,m,f,formatting)**
  - Formatting is the same as with the ON operator
- **BTITLE**
  - String to be used for the break title
- **BTOTAL and BAVERAGE**
  - Section statistics at the end of each section

30 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)


© 2012, 2013 IBM Corporation



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) -**  
 ↳ ddnames of data sets

**TITLE('BOOKS FOR INDIVIDUAL PUBLISHERS') PAGE -**  
 ↳ Title line elements

**HEADER('TITLE OF BOOK') ON(1,35,CH) -**  
 ↳ Heading and field

**HEADER('PRICE OF BOOK') ON(170,4,BI,C1,F'\$') -**  
 ↳ Heading and field


**BTITLE('PUBLISHER:') BREAK(106,4,CH) -**  
 ↳ Break field  
 ↳ Break title

**BAVERAGE('AVERAGE FOR THIS PUBLISHER') -**  
 ↳ Section average

**BTOTAL('TOTAL FOR THIS PUBLISHER') -**  
 ↳ Section total

**AVERAGE('AVERAGE FOR ALL PUBLISHERS') -**  
 ↳ Overall average

**TOTAL('TOTAL FOR ALL PUBLISHERS') -**  
 ↳ Overall total

31 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)      © 2012, 2013 IBM Corporation      

For this example, we will use the data set with books from publishers VALD and WETH, sorted by publisher and title, that we created previously. To print a report with sections by publisher showing the title and price fields with a title line, field headings, break title, break averages and totals, and overall averages and totals.

## Sectioned Report Output



BOOKS FOR INDIVIDUAL PUBLISHERS - 1 -	
PUBLISHER: VALD	
TITLE OF BOOK	PRICE OF BOOK
CELLS AND HOW THEY WORK	\$24.95
. . .	. . .
ZEN BUSINESS	\$12.00
AVERAGE FOR THIS PUBLISHER	\$17.91
TOTAL FOR THIS PUBLISHER	\$179.14

BOOKS FOR INDIVIDUAL PUBLISHERS - 2 -	
PUBLISHER: WETH	
TITLE OF BOOK	PRICE OF BOOK
ANTICIPATING THE MARKET	\$20.00
. . .	. . .
THE INDUSTRIAL REVOLUTION	\$7.95
AVERAGE FOR THIS PUBLISHER	\$18.53
TOTAL FOR THIS PUBLISHER	\$166.77

32 complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



The output would be like shown above. I put everything into a single slide so that you can see the complete output in a single slide. In this case the output shows the books in stock for the publisher VALD, and provides an average price and the sum of the inventory. In this case the output shows the books in stock for the publisher WETH, and provides an average price and the sum of the inventory. This output shows the average for both publishers and the sum of the inventory.





## Sectioned Report Output (continued)

BOOKS FOR INDIVIDUAL PUBLISHERS	
TITLE OF BOOK	PRICE OF BOOK
-----	
AVERAGE FOR ALL PUBLISHERS	\$18.20
TOTAL FOR ALL PUBLISHERS	\$345.91

## How Many Times Fields Occur



- **OCCUR** operator how many times ON fields occurs
  - **ALLDUPS** - Duplicate values
  - **NODUPS** - Non-duplicate values
  - **EQUAL** - Specified number of times
  - **HIGHER** - More than a specified number of times
  - **LOWER** - Less than a specified number of times
  - **ON(VALCNT)** - Each field value occurs
  - **ON(VLEN)** - Length of VLR records

### **OCCURS report is similar to DISPLAY report**

**ALLDUPS** - only list duplicate values

**NODUPS** - only list non-duplicate values

**EQUAL** - only list values that occur a specified number of times

**HIGHER** - only list values that occur more than a specified number of times

**LOWER** - only list values that occur less than a specified number of times

**ON(VALCNT)** can be used to print the number of times each field value occurs.

**ON(VLEN)** can be used for the record length of variable length records.

## How Many Times Fields Occur

\* Print the count of books in use from each publisher

OCCUR FROM(BKIN) LIST(PUBCT) BLANK -

→ Alternate print format

→ ddnames of data sets

TITLE('Books from Publishers') DATE(DMY.) -

→ Title line elements

HEADER('Publisher') HEADER('Books Used') -

→ Field headings

ON(106,4,CH) ON(VALCNT,N05)

→ Publisher and Count

For this example, we will use the data set with books from publishers VALD and WETH, sorted by publisher and title, that we created previously. To print a report with sections by publisher showing the title and price fields with a title line, field headings, break title, break averages and totals, and overall averages and totals.

## How Many Times Fields Occur Output



Books from Publishers 11.01.13

Publisher Books Used

-----	
COR	7
FERN	4
VALD	5
WETH	4

This shows the number of occurrences for each publisher.

## Records by Field Occurrences



- **SELECT** operand to create an output dataset with how many times different ON field values occur
  - **FIRST** - First record for each value
  - **LAST** - Last record for each value
  - **FIRSTDUP** - First record for duplicate values
  - **LASTDUP** - Last record for duplicate values
  - **ALLDUPS** - All records with duplicate values
  - **NODUPS** - Records with non-duplicate values
  - **EQUAL** - Specified number of times
  - **HIGHER** - More than a specified number of times
  - **LOWER** - Less than a specified number of times

37 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



You can use ICETOOL's SELECT operator to create an output data set with records selected according to how many times different ON field values occur, sorted by those ON field values. As with the OCCUR operator, values that occur only once are called non-duplicate values, and values that occur more than once are called duplicate values.

**FIRST** - keep only the first record for each value (that is, records with non-duplicate values, and the first record for duplicate values)

**LAST** - keep only the last record for each value (that is, records with non-duplicate values, and the last record for duplicate values)

**FIRSTDUP** - only keep the first record for duplicate values

**LASTDUP** - only keep the last record for duplicate values

**ALLDUPS** - only keep records with duplicate values

**NODUPS** - only keep records with non-duplicate values

**EQUAL** - only keep records with values that occur a specified number of times

**HIGHER** - only keep records with values that occur more than a specified number of times

**LOWER** - only keep records with values that occur less than a specified number of times

## Records by Field Occurrences

\* Course name and author's last name for courses with more than one book  
 \* and only one book  
**SELECT FROM(BKIN) TO(DUP) DISCARD(NODUP)**  
 → ddname of input data set  
 → ddname of dup output data set  
 → ddname of nodup output data set

**ON(120,25,CH) ALLDUPS USING(CTL1)**  
 → First four chars of DFSORT control data set  
 → Criteria for selecting records  
 → Course name

### AllDups

INTRO TO COMPUTERS	CHATTERJEE
INTRO TO COMPUTERS	CHATTERJEE
INTRO TO COMPUTERS	CHATTERJEE
MODERN POETRY	FRIEDMAN
MODERN POETRY	FRIEDMAN
WORLD HISTORY	GOODGOLD
WORLD HISTORY	WILLERTON

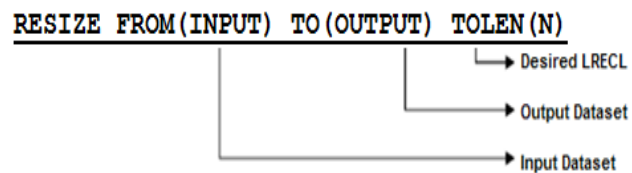
### NoDups

ADVANCED MARKETING	LORCH
BIOLOGY I	GREENBERG
DATA MANAGEMENT	SMITH
EUROPEAN HISTORY	BISCARDI
FICTION WRITING	BUCK
MARKETING	MAXWELL
PSYCHOANALYSIS	NAKATSU
PSYCHOLOGY I	ZABOSKI
TECHINCAL EDITING	MADRID

These two publishers have more than 4 books in the inventory.

## Create small records from large records and vice versa

- **RESIZE** operator to create multiple smaller fixed-length records from a larger fixed-length record or to create one long fixed-length record from several shorter fixed-length records.



Sometimes you may want to split a large record into smaller records or combine many small records into one large record. You can use ICETOOL's RESIZE operator to build short records from large records or vice versa.

**FROM(indd)** - would be the input dataset

**TO(outdd)** - would be the output dataset

**TOLEN(n)** - Specifies the record length you want ICETOOL to use for the resized output records. n can be 1 to 32760. n must not be equal to the input record length.

## Creating Small records from Large record

Let's say we have an input file with RECFM=FB and LRECL=52 that has these records:

```
-----1-----2-----3-----4-----5--  
*SECTION 001**SECTION 002**SECTION 003**SECTION 004*  
*SECTION 005**SECTION 006**SECTION 007**SECTION 008*
```

```
//TOOLIN DD *  
  RESIZE FROM(INPUT) TO(OUTPUT) TOLEN(13)  
//*
```

We want to split each 52-byte record into four 13-byte records. The output data set will have RECFM=FB and LRECL=13 and contain these records:

```
*SECTION 001*  
*SECTION 002*  
*SECTION 003*  
*SECTION 004*  
*SECTION 005*  
*SECTION 006*  
*SECTION 007*  
*SECTION 008*
```

As you can see we took 2 records each with a length of 52 bytes and created 8 records each with 13 bytes length.



## Creating Large Records From Small Records

Lets say we have an input file with RECFM=FB and LRECL=13 that has these records:

```
-----1-----2-----3-----4-----5--
*SECTION 001*
*SECTION 002*
*SECTION 003*
*SECTION 004*
*SECTION 005*
*SECTION 006*
*SECTION 007*
*SECTION 008*
```

```
//TOOLIN DD *
  RESIZE FROM(INPUT) TO(OUTPUT) TOLEN(52)
//*
```

We want to combine 4 13-byte records into a single 52-byte record. The output data set will have RECFM=FB and LRECL=52 and contain these records:

```
-----1-----2-----3-----4-----5--
*SECTION 001**SECTION 002**SECTION 003**SECTION 004*
*SECTION 005**SECTION 006**SECTION 007**SECTION 008*
```

Now that you have seen how a large record is broken into small record, lets do the reverse. As you can see all we need is to change the TOLEN field to 52 and voila we created 2 records combining four thirteen byte records into a single record. This might come handy when you want combine all the continuation messages (ex: DSNT501I or DSNT501I) from your syslog. Remember that RESIZE only works on fixed records.

## Defining and Using Symbols



- **SYMNAMES DD**
  - RECFM=FB and LRECL=80
- Used in any ICETOOL or DFSORT JOB
  
- **TITLE**
  - Name of the Symbol created
  - 1 – 50 characters
  - Letters, numbers, \$, @, \_, and –
  - First character NOT a number
  - Provide length and format
  - SYMBOL, Symbol, and symbol
    - 3 different symbols

42 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



DFSORT and ICETOOL obtain the symbols to be used from the data set specified in a SYMNAMES DD statement. Create the SYMNAMES data set you want to use with RECFM=FB and LRECL=80 in the same way you would create a data set containing DFSORT JCL and control statements. Then use an editor, such as ISPF EDIT, to write the SYMNAMES statements defining your symbols.

This is the symbol you will use for the Title field. A symbol can be 1 to 50 characters consisting of uppercase letters(A-Z), lowercase letters (a-z), numbers (0-9), the number sign (#), the dollar sign (\$), the commercial at sign (@), the underscore(\_), and the hyphen (-). However, the first character must not be a number. Title, TITLE, and title are three different symbols.

## Defining and Using Symbols



```
• Author_First_Name,*,15,CH
• Publisher,*,4,CH
• Instructor_Last_Name,*,15,CH
• Instructor_Initials,*,2,CH
• Price,*,4,BI
• ...
//IN          DD DSN=A123456.SORT.SAMPIN,DISP=SHR
//SYMNAMES DD DSN=A123456.SORT.SYMBOLS,DISP=SHR
//OUT        DD DSN=A123456.SORT.SAMPOUT,DISP=OLD
//SYMNOUT DD SYSOUT=*
//TOOLIN DD *
SORT FROM(IN) TO(OUT) USING(CTL1)
RANGE FROM(OUT) ON(Price) LOWER(+700)
RANGE FROM(OUT) ON(Price) HIGHER(+2000)
//CTL1CNTL DD *
SORT FIELDS=(Instructor_Last_Name,A,Instructor_Initials,A,
              Price,D)
/*
```

43 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



You will notice that the SYMBOLES can be used in either the TOOLIN or Control statement area.

The SYMNAMES DD statement specifies the SYMNAMES data set to be used for this application. The SYMNOUT DD statement specifies a data set in which you want DFSORT to list your original SYMNAMES statements and the symbol table constructed from them. You can omit the SYMNOUT data set if you don't want to see that information.

## Defining and Using Symbols



- Use Symbols to define constants
  - Decimal, character, hexadecimal constants or bit
- From an earlier slide
  - RANGE FROM(OUT) ON(Price) LOWER(+700)
  - RANGE FROM(OUT) ON(Price) HIGHER(+2000)
- If you define
  - Discount,+700
  - Premium,+2000
- You could use
  - RANGE FROM(OUT) ON(Price) LOWER(Discount)
  - RANGE FROM(OUT) ON(Price) HIGHER(Premium)

44 Complete your sessions evaluation online at [SHARE.org/SanFranciscoEval](http://SHARE.org/SanFranciscoEval)

© 2012, 2013 IBM Corporation



You can use symbols wherever decimal constants, character constants, hexadecimal constants or bit constants can appear in DFSORT control statements and ICETOOL operators.

## REFERENCES.

- The DFSORT home page on the World Wide Web at URL: <http://www.ibm.com/storage/dfsor>
- DFSORT Application Programming Guide (SC26-7523)
- DFSORT: Getting Started (SC26-7527)
- z/OS DFSMSrmm Reporting (SC26-7406)
- z/OS DFSMSHsm Data Recovery Scenarios (GC35-0419)
- z/OS SecureWay Security Server RACF Auditor's Guide (SA22-7684)
- z/OS SecureWay Security Server RACF Security Administrator's Guide (SA22-7683)

## REFERENCES.

- RACFICE2 describes a technique for analyzing RACF data using ICETOOL. You can obtain RACFICE2 at: [www.ibm.com/systems/z/os/zos/features/racf/downloads/racfice.html](http://www.ibm.com/systems/z/os/zos/features/racf/downloads/racfice.html)
- The DFSORT product tape contains a set of illustrative examples of interest to Storage Administrators and others who analyze data created by DFHSM, DFSMSrmm, DCOLLECT and SMF. The source for the following examples are available in sample job ICESTGEX:
  - DCOLEX1 - DCOLLECT Ex 1: VSAM report
  - DCOLEX2 - DCOLLECT Ex 2: Conversion reports

## REFERENCES.

- DCOLEX3 - DCOLLECT Example 3: Capacity planning analysis and reports.
- DFHSMEX1 - DFHSM Example 1: Deciphering Activity Logs
- DFHSMEX2 - DFHSM Example 2: Recover a DFHSM CDS with a broken index.
- RMMEX1 - DFSMSrmm Example 1: SMF audit report.
- RMMEX2 - DFSMSrmm Example 2: Create ADDVOLUME commands.
- ICESTGEX is also available via anonymous FTP from:  
[ftp.software.ibm.com/storage/dfsrmv/](ftp://software.ibm.com/storage/dfsrmv/)

#SHAREorg



# DFSMS DFSORT: The ICETOOL Cometh - Getting Started Using ICETOOL

Sri Hari Kolusu  
IBM

Wednesday, February 6<sup>th</sup> 2013  
Session Number : 12983

Skolusu@us.ibm.com

