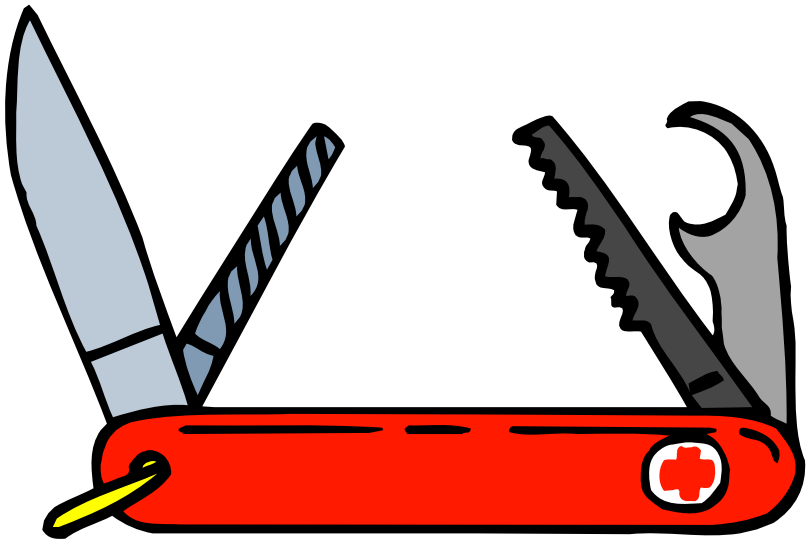
DFSORT and ICETOOL

KOIRUG - May 2007



Robert S. Hansel

RACF Specialist - RSH Consulting, Inc.

r.hansel@rshconsulting.com - 617-969-9050 - www.rshconsulting.com

TOPICS

Introduction to the DFSORT Program Product

DFSORT

- How to execute DFSORT
- DFSORT Program Control Statements
- DFSORT Examples

ICETOOL

- How to execute ICETOOL
- ICETOOL Program Control Statements
- ICETOOL Examples

Hints and Tips

References

INTRODUCTION

"DFSORT is IBM's high-performance sort, merge, copy, and analysis reporting product for z/OS."

DFSORT is a licensed program and is part of the IBM Data Facility family of products

What functions can you use DFSORT for?

Sort	**Change the sequence of data**
Copy	**Duplicate data**
Merge	**Combine multiple sources of data**
Expand	**Create multiple files from one input file**
Filter	**Select subsets of data**
Reformat	**Create new records from existing data**

INTRODUCTION

"ICETOOL is a multipurpose DFSORT utility that uses the capabilities of DFSORT to perform multiple operations on one or more data sets in a single step."

ICETOOL is a component of the DFSORT product

What is ICETOOL used for?

- **Generating Reports**
- **Statistical Analysis**
- **Performing multiple DFSORT functions in a single job step**

DFSORT

DFSORT EXECUTION

How is DFSORT Invoked ?

- With an EXEC job control statement in the input stream using the name of the program or the name of a cataloged procedure
- Called from a compiled program written in Assembler, COBOL, PL/I or REXX
- Called from a CLIST or REXX exec under TSO
- Called by the ICETOOL utility
- Through interactive panels supported under ISPF and ISMF

DFSORT EXECUTION

Basic DD Statements

- **SORTIN** — **Input File containing records to be processed**
- **SORTOUT** — **Output File to be created by DFSORT**
- **SYSIN** — **Specify most DFSORT control statements**
- **DFSPARM** — **Specify certain overrides and all control statements**
- **SYSOUT** — **DFSORT Processing Messages**
- **SORTWK*nn*** — **Optional intermediate storage datasets for a sort**
- ***outfil*** — **OUTFIL output datasets which replace SORTOUT**

DFSORT EXECUTION

Sample JCL for invoking DFSORT:

```
//jobname  JOB  Statement
…
//STEP1    EXEC PGM=SORT,REGION=100M
//SORTIN   DD DSN=MY.RACF.IRRDBU00.UNLOAD,DISP=SHR
//SORTOUT  DD DSN=MY.OUTPUT.FILE,DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,LRECL=4096,BLKSIZE=0)
//SYSIN    DD *
 DFSORT Program Control Statements
//SYSOUT   DD SYSOUT=*
//DFSPARM  DD *
 DFSORT Option Overrides and Program Control Statements
```

DFSORT Control Statements

Most Commonly Used DFSORT Program Control Statements

SORT	INCLUDE
OPTION	OUTFIL

Less Commonly Used

ALTSEQ	DEBUG
END	INREC
MERGE	OMIT
OUTREC	MODS
RECORD	SUM

DFSORT Control Statements

In order to code control statements you must know the structure of SORTIN file

- **Record layout as to location and length of data fields**
- **Format of any fields you need to reference (e.g., character, binary)**
- **DCB RECFM – Fixed or Variable**
  - **Variable length records are prefixed by a 4-byte length field; when working with variable length records, add 4 to field starting positions**

Basic control statement syntax

- **Statements must be coded in columns 2 to 72**
- **Any statement ending in comma, semicolon, or colon implies a continuation, as does a non-blank character in column 72**
- **A line with an asterisk (*) in column 1 is treated as a comment**
- **Blank lines are ignored**

DFSORT EXAMPLE

EXAMPLE: Create a file with all members of RACF group SYS1 sequenced with the most recent connection dates to SYS1 first

```
//SORTIN   DD DSN=MY.IRRDBU00.RACF.UNLOAD,DISP=SHR
//SORTOUT  DD DSN=MY.USERIDS.SORTED,DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE),
//            DCB=(RECFM=VB,LRECL=4096,BLKSIZE=0)
//DFSPARM  DD *
 OPTION VLSHRT
//SYSIN    DD *
 SORT FIELDS=(57,10,CH,D,10,8,CH,A)
 INCLUDE COND=(5,4,CH,EQ,C'0205',AND,
               19,8,CH,EQ,'SYS1    ')
 OUTREC FIELDS=(1,4,57,10,15:X'40',10,8)
//SYSOUT   DD SYSOUT=*
```

DFSORT Control Statements

DFSORT requires at least one control statement to specify the operation that the user wishes performed

- **Sort Operation** — **SORT statement**
- **Copy Operation** — **SORT FIELDS=COPY statement**
- **Merge Operation** — **MERGE statement**

Only one operation may be performed in each execution of DFSORT

Records can be reformatted both before and after sorting using INREC and OUTREC

DFSORT Control Statements - SORT

SORT Statement - controls the sequence of output records

SORT FIELDS=(Control-Field1 [, … ,Control-Field*n*])
[,FORMAT=*format*]

Control-Field: start,length,format,order (e.g., 110,5,CH,A)

Format: 30 possible formats (e.g., CH - Character)

Order: Ascending | Descending

If all fields are the same format, a single FORMAT operand can be used in lieu of coding the format in each Control Field

Combined length of all sort fields must not exceed 4092 bytes

DFSORT Control Statements - SORT

SORT Statement Examples:

```
SORT FIELDS=(57,10,CH,D,10,8,CH,A)
SORT FIELDS=(57,10,D,10,8,A),FORMAT=CH
```

Control Field 1:

- Starts in position 57
- Contains 10 bytes of character data
- Is to be sorted in descending sequence

Control Field 2:

- Starts in position 10
- Contains 8 bytes of character data
- Is to be sorted in ascending sequence

DFSORT Control Statements - SORT

Consider an input file containing RACF IRRDBU00 Unload Type 0205 User Connection records for Group SYS1:

Record Type	**Start=1,Length=4**
Userid	**Start=6,Length=8**
Group	**Start=15,Length=8**
. . .	
Last-Connect-Date	**Start=53,Length=10**

Example: Sort the records by Userid in ascending sequence within connection date in descending sequence

```
SORT FIELDS=(57,10,D,10,8,A),FORMAT=CH
```

DFSORT Control Statements - COPY

COPY Statement - requested by specifying the FIELDS=COPY operand with the SORT statement

```
SORT FIELDS=COPY
```

Other control statements which are commonly used in conjunction with a copy operation are:

- **INCLUDE**
- **OUTFIL**
- **OUTREC**

DFSORT Control Statements

Record selection statements

INCLUDE	**Specifies that only records whose fields meet certain criteria are included**
OMIT	**Specifies that any records whose fields meet certain criteria are excluded**
OUTFIL	**Specifies the subsets of records to be included or omitted in multiple output data sets**

DFSORT Control Statements - INCLUDE

INCLUDE statement

INCLUDE COND=(Logical-Test1 [,AND | OR, … ,Logical-Test*n*])
[,FORMAT=*format*]

Types of Logical Tests

- **Comparison Test**
- **Substring Comparison Test**
- **Bit Logic**
- **Numeric**
- **Date**

When comparing a field to a constant, pad the constant to ensure it is the same length as the field (e.g., C'SYS1 ')

DFSORT Control Statements - INCLUDE

INCLUDE Statement - Comparison Test

Start,Length,Format,Test,Test-Value

Test:	**EQ**	**Equal**
	NE	**Not Equal**
	GT	**Greater Than**
	GE	**Greater Than or Equal To**
	LT	**Less Than**
	LE	**Less Than or Equal To**

Test-value: **string | start,length,format**

String: **Character - C'*value*'** **Hexadecimal - X'*hex-value*'**

Examples: **5,4,CH,EQ,C'0200'**

10,8,CH,EQ,30,8,CH

DFSORT Control Statements - INCLUDE

INCLUDE Statement - Substring (SS) Comparison Test

- Find a string within a field
- Find field contents within a set of strings

Start,Length,SS,Test,Test-Value[s]

Test: EQ, NE

Separate multiple Test-Values with a character that is not likely to appear in the field to be tested (e.g., comma)

Examples: INCLUDE COND=(10,44,SS,EQ,C'*')

INCLUDE COND=(19,8,SS,EQ,C'SYS1 ,SYS2 ')

INCLUDE COND=(5,4,SS,NE,C'0205,0203,0200')

DFSORT Control Statements - INCLUDE

INCLUDE Statement - AND and OR

- AND - all conditions must be met
- OR - any one of several conditions must be met
- ANDs are evaluated before ORs
- Expressions inside parentheses are evaluated first
- The number of tests is limited only by available storage
- Characters '&' and '|' can be used for 'AND' and 'OR' respectively

```
Example:    INCLUDE COND=(5,4,CH,EQ,C'0205',AND,
                         (19,8,CH,EQ,'SYS1    ',OR,
                          19,8,CH,EQ,'SYS2    '),AND,
                         57,10,CH,GE,C'2006-01-00')
```

DFSORT Control Statements - OUTFIL

OUTFIL Statement - instructs DFSORT to write selected records to output files other than SORTOUT

OUTFIL FNAMES=*ddname*,Operand1[,Operand2, . . . ,Operand*n*]

Operands: INCLUDE | OMIT | SAVE | …

Processing Considerations:

- **Use of OUTFIL overrides default of SORTOUT as an output file**
- **Multiple OUTFIL statements are allowed**
- **Each OUTFIL can have different filtering specifications**
- **FILES=*ddname* can be used instead of FNAMES=*ddname***
- **A DD statement *ddname* is required for each FNAMES= | FILES=**
- **SAVE collects any records not selected by other OUTFIL statements**

DFSORT Control Statements - OUTFIL

OUTFIL example: Create three output files containing:

- **All 0100 Group Records**
- **All Userid (0200) and Connection (0205) records**
- **All other records**

```
//SORTIN      DD      DSN=RACF.UNLD,DISP=SHR
//OUTDD1      DD      DSN=RACF.UNLD.R100.RECS,DISP=…
//OUTDD2      DD      DSN=RACF.UNLD.R200#205.RECS,DISP=…
//OUTDD3      DD      DSN=RACF.UNLD.EVERY.THING.ELSE,DISP=…
//SYSIN       DD      *
 SORT FIELDS=COPY
 OUTFIL FNAMES=OUTDD1,INCLUDE=(5,4,CH,EQ,C'0100')
 OUTFIL FNAMES=OUTDD2,INCLUDE=(5,4,SS,EQ,C'0200,0205')
 OUTFIL FNAMES=OUTDD3,SAVE
```

DFSORT Control Statements - OPTION

OPTION statement - directs DFSORT to use program options which override installation defaults

OPTION Option1 [,Option2, … ,Option*n*]

Considerations:

- **There are over 35 different options which may be specified**
- **Certain Options can only be specified via the DFSPARM DD statement**
- **Only necessary when you want to override installation defaults**
  - **SYSOUT DD output OPTIONS: lines list the installation defaults**

DFSORT Control Statements - OPTION

Commonly used DFSORT OPTIONs:

VLSHRT	**Continue processing if a variable length record is too short to contain one of the sort fields or one of the OMIT/INCLUDE fields**
VLSCMP	**Pad short OMIT/INCLUDE fields with binary zeroes**
SORTIN=*ddname*	**Use a DDNAME other than SORTIN as the input file for a SORT or COPY operation**
DYNALLOC=(*d,n*)	**Dynamically allocate *n SORTWKnn* work datasets on DASD devices *d* (e.g., DYNALLOC=(SYSDA,4))**
EQUALS	**Maintain the existing sequence of records when control fields are equal**
SKIPREC=*n*	**Skip first *n* records before starting processing**
STOPAFT=*n*	**Read only the first *n* records from input (useful in testing)**

DFSORT Control Statements

Less Commonly Used DFSORT Control Statements

ALTSEQ	**Change collating sequence**
DEBUG	**Problem diagnosis**
END	**Stop reading control statements**
INREC	**Reformat records before sorting**
MERGE	**Combine similar files**
OMIT	**Mutually exclusive with INCLUDE**
OUTREC	**Reformat records after sorting**
MODS	**Specify User Exits**
RECORD	**Describe type and length of records**
SUM	**Total fields on equal records**

DFSORT Constraints

Must know record layouts

Must add 4 to start positions for variable length record fields

Only one operation per job step

Lack of reporting facilities

Limited statistical information

ICETOOL

ICETOOL

How is ICETOOL Invoked ?

- With an EXEC job control statement in the input stream using the name of the program or the name of a cataloged procedure
- Called from a compiled program written in Assembler, COBOL, PL/I or REXX
- Called from a CLIST or REXX exec under TSO

ICETOOL EXECUTION

Basic DD Statements

- **TOOLIN** — **ICETOOL Control Statements**
- **TOOLMSG** — **ICETOOL Messages**
- ***xxxx*CNTL** — **DFSORT Control Statements**
- **DFSMSG** — **DFSORT Processing Messages**
- ***indd*** — **Input data set**
- ***outdd*** — **Output data set for COPY, SELECT, or SORT**
- ***savedd*** — **Output dataset for SELECT**
- ***listdd*** — **List output dataset for DEFAULTS, DISPLAY, or OCCUR**

*xxxx*CNTL in ICETOOL is used like DFSPARM with DFSORT - to pass control statements to DFSORT

- **Multiple *xxxx*CNTL DD statements are allowed, enabling multiple sort operations to be performed in one ICETOOL job step**

ICETOOL EXECUTION

Sample JCL for invoking ICETOOL:

```
//jobname  JOB  Statement
…
//STEP1    EXEC PGM=ICETOOL,REGION=300M
//TOOLIN   DD *
 ICETOOL Control Statements
//TOOLMSG  DD SYSOUT=*
//DFSMSG   DD SYSOUT=*
//xxxxCNTL DD *
 DFSORT Option Overrides and Program Control Statements
//indd     DD DSN=RACF.UNLOAD,DISP=SHR
//outdd    DD DSN=INTERIM.COPY.SORT.OUTPUT,DISP=…
//listdd   DD DSN=REPORT.OUTPUT,DISP=(NEW,CATLG,DELETE),…
```

ICETOOL Control Statements

Commonly Used ICETOOL Operators

COPY	SORT	Record selection
DISPLAY	OCCUR	Report generation
MODE	COUNT	Processing control

Other ICETOOL Operators

DEFAULTS	RANGE
SELECT	SPLICE
STATS	UNIQUE
VERIFY	

Basic control statement syntax

- Statements must be coded in columns 1 to 72
- Continuation can be indicated by a hyphen (-) after the operator or after any operand

ICETOOL Control Statements - COPY

COPY operator - copy an input dataset to one or more output datasets (up to 10); each output dataset can be built with different DFSORT specifications

COPY FROM(*indd*) [USING(*xxxx*)] [TO(*outdd*1 [, … ,*outdd*10])]

Optionally provide USING along with COPY specifications in *xxxx*CNTL DD statement

Must include either USING or TO to specify output files

- **USING and TO may be used in combination**
- **If USING not specified, all records are copied**

Can specify output datasets by either:

- **TO(*outdd*) on COPY operator**
- **OUTFIL FNAMES=*outdd* in *xxxx*CNTL statements**

ICETOOL Control Statements - COPY

COPY Examples: Select various unload record types

```
//STEPICE  EXEC PGM=ICETOOL,REGION=100M
…
//RACFDB   DD DSN=RLW.R16DBUA,DISP=SHR
//USERDD   DD DSN=RLW.SORTOUT,DISP=SHR
//TOOLIN   DD *
  COPY FROM(RACFDB) USING(C400) TO(R0400)
  COPY FROM(RACFDB) USING(CSEL)
//C400CNTL   DD *
  INCLUDE COND=(5,4,CH,EQ,C'0400')
  OPTION VLSHRT
//CSELCNTL   DD *
  OUTFIL FNAMES=R0500,INCLUDE=(5,4,CH,EQ,C'0500')
  OUTFIL FNAMES=R0505,INCLUDE=(5,4,CH,EQ,C'0505')
  OPTION VLSHRT
//R0400      DD DSN=RLW.R0400RCD,DISP=…
//R0500      DD DSN=RLW.R0500RCD,DISP=…
//R0505      DD DSN=RLW.R0505RCD,DISP=…
```

ICETOOL Control Statements - SORT

SORT operator - sorts an input dataset to one or more output datasets

SORT FROM(*indd*) USING(*xxxx*) [TO(*outdd*)]

USING is required and must provide SORT specifications in *xxxx*CNTL DD statement

Can specify output dataset by either:

- **TO(*outdd*) on SORT operator**
- **OUTFIL FNAMES=*outdd* in *xxxx*CNTL statements**

ICETOOL Control Statements - SORT

SORT Example: Select 0205 User Connect records and sort by USERID within Group

```
//STEPICE  EXEC PGM=ICETOOL,REGION=100M
…
//RACFDB   DD DSN=RLW.R16DBUA,DISP=SHR
//USERDD   DD DSN=RLW.SORTOUT,DISP=SHR
//TOOLIN   DD *
  SORT FROM(RACFDB) USING(SORT)
//SORTCNTL DD *
  SORT FIELDS=(19,8,CH,A,10,8,CH,A)
  INCLUDE COND=(5,4,CH,EQ,C'0205')
  OUTFIL FNAMES=USERDD
```

ICETOOL Control Statements - DISPLAY

DISPLAY operator - prints the values of specified fields to a separate dataset

```
DISPLAY   FROM(indd) LIST(listdd) -
          [ PAGE TITLE('string') DATE[(format)] TIME[(format)] - ]
          [ BLANK - ]
          ON(start,length,format) [ - ]
          [ HEADER('string'  |  NONE) - ]
```

PAGE	**Print page number, automatically formatted as: - # -**
BLANK	**Prints numerics right justified, no leading 0s or + sign**
ON	**Report data column (up to 20)**
HEADER	**Column heading (up to 50 characters)**

The ON and HEADER statements are paired in the sequence coded

The maximum output line length is 2048 bytes

ICETOOL Control Statements - DISPLAY

DISPLAY Example: Print Userid, User Name, and Last Logon Date from an extract of RACF Database Unload 0200 records

```
//STEPICE  EXEC PGM=ICETOOL
…….
//RACFDB   DD DSN=YOUR.RACF.UNLOAD,DISP=SHR
//USERDD   DD DSN=YOUR.RACF.USERIDS,DISP=SHR
//LISTOUT  DD SYSOUT=*
//TOOLIN   DD *
COPY FROM(RACFDB) TO(USERDD) USING(ICE1)
DISPLAY FROM(USERDD) LIST(LISTOUT) –
PAGE TITLE('LAST LOGON REPORT') DATE TIME -
HEADER('USERID') ON(10,8,CH) -
HEADER('USER NAME') ON(79,20,CH) -
HEADER('LAST LOGON') ON(118,10,CH)
//ICE1CNTL     DD *
 INCLUDE COND=(5,4,CH,EQ,C’0200’)
 OPTION VLSHRT
```

ICETOOL Control Statements - DISPLAY

```
- 1 -        LAST LOGON REPORT          04/20/07        08:24:41

USERID           USER NAME                LAST LOGON
-------------    --------------------     ------------------
$OEDFLU          OE-USS DEFAULT USER
irrcerta         CERTAUTH Anchor
irrmulti         Criteria Anchor
irrsitec         SITE Anchor
ANONYMO          ####################     2005-06-02
BPXOINIT         BPXOINIT
CICSUSER         CICSUSER                 2004-04-08
CLRLOG           LOGREC CLEAR             2006-05-15
DSN1WLM1         ####################
FTPD             FTPD                     2006-12-18
IBMUSER                                   2006-10-05
INETD            INETD
INTERNAL         ####################
```

ICETOOL Control Statements - OCCUR

OCCUR operator - Prints each unique value for specified numeric or character fields and how many times it occurs

```
OCCUR   FROM(indd) LIST(listdd) -
        [ PAGE TITLE('string')  DATE[(format)] TIME[(format)] - ]
        [ BLANK - ]
        ON(start,length,format  |  VALCNT) [ - ]
        [ HEADER('string'  |  NONE) - ]
        [ HIGHER(n) ]
```

ON	Report data columns (up to 10)
ON(VALCNT)	Display count of instances of occurrence
HIGHER(*n*)	Only list those instances with a count higher than *n*

Combination of all ON data items is used as an instance for counting

ICETOOL Control Statements - OCCUR

OCCUR Example - List CICS OPIDENTs assigned to more than 1 USERID, and a count of how many for each OPIDENT

```
//STEPICE  EXEC PGM=ICETOOL
…
//RACFDB   DD DSN=YOUR.RACF.UNLOAD,DISP=SHR
//USERDD   DD DSN=&&SORT,DISP=(NEW,DELETE,DELETE),…
//REPORT   DD SYSOUT=*
//TOOLIN   DD *
SORT FROM(RACFDB) TO(USERDD) USING(OPID)
OCCUR FROM(USERDD) LIST(REPORT) –
DATE TIME PAGE TITLE('SHARED CICS OPIDENTS') -
BLANK -
HEADER('OPID') ON(19,3,CH) -
HEADER('# USERS') ON(VALCNT) -
HIGHER(1)
//OPIDCNTL     DD *
 SORT FIELDS=(19,3,CH,A)
 INCLUDE COND=(5,4,CH,EQ,C’0230’,AND,19,3,CH,NE,C'   ')
 OPTION VLSHRT
```

ICETOOL Control Statements - OCCUR

```
05/20/07        09:00:24        - 1 -        SHARED CICS OPIDENTS

OPID             # USERS
----    ---------------
C01                    2
PAY                    4
WWW                   18
```

ICETOOL Control Statements - MODE

MODE operator - controls ICETOOL error checking and processing after error detection

MODE STOP | CONTINUE | SCAN

STOP	**(Default) Discontinue processing if a return code of 12 or 16 is returned**
CONTINUE	**Continue processing even if a return code of 12 or 16 is returned**
SCAN	**Scan the statements for errors but not to invoke DFSORT**

MODE can be specified multiple times during ICETOOL execution to govern processing of subsequent operators

ICETOOL Control Statements - COUNT

COUNT operator - print a message in the TOOLMSG DD statement output showing the number of records, or subset of records, on the file

COUNT FROM(*indd*) [USING(*xxxx*)]

Optionally provide USING along with record selection criteria in *xxxx*CNTL DD statement to count subset of records

```
COUNT FROM(indd) USING(xxxx)
//xxxxCNTL  DD *
 INCLUDE COND=(…)
```

ICETOOL Control Statements - COUNT

The COUNT value can optionally be tested against certain conditions with the result used to set a return code (often used in conjunction with MODE)

COUNT FROM(*indd*) *Test-Condition* [RC4] [USING(*xxxx*)]

Test Conditions (*n* = numeric count value):

EMPTY	**NOTEMPTY**
HIGHER*(n)*	**LOWER(*n*)**
EQUAL(*n*)	**NOTEQUAL(*n*)**

If the test condition is met, ICETOOL sets RC=12; otherwise, RC is set to 0; whereas, the RC4 operand converts RC12 to RC4

ICETOOL Control Statements - COUNT

COUNT Example: Cease further processing if the input dataset is empty or have the job complete with a Return Code of 4 if the input dataset has more than 10 users with UID of 0

```
//STEPICE  EXEC PGM=ICETOOL
//INDD     DD DSN=input,DISP=SHR
//TOOLIN   DD *
 MODE STOP
 COUNT FROM(INDD) EMPTY
 COUNT FROM(INDD) HIGHER(10) USING(UID0) RC4
 SORT  FROM(INDD) USING(UID0) TO(TEMPDD)
 DISPLAY …
//UID0CNTL DD *
 INCLUDE COND=(5,4,CH,EQ,C'0270',AND,19,10,CH,EQ,C'0000000000')
 OPTION VLSHRT
//TEMPDD   DD …
```

ICETOOL Control Statements

OTHER ICETOOL Operators

DEFAULTS	**Prints DFSORT installation defaults**
RANGE	**Prints count of values in a specific range for a numeric field**
SELECT	**Filters records based on presence of fields**
SPLICE	**Combine contents of multiple input records**
STATS	**Prints statistics for specified numeric fields**
UNIQUE	**Prints count of unique values for character or numeric field**
VERIFY	**Identifies invalid decimal fields**

ICETOOL Problem Example

Problem: Identify and report audit status of all dataset profiles with a LEVEL field other than 000.

IRRDBU00 0400 Record

DSBD_RECORD_TYPE	(1,4)	= '0400'
DSBD_AUDIT_LEVEL	(143,8)	
DSBD_LEVEL	(170,3)	Non-zero
DSBD_AUDIT_OKQUAL	(448,8)	
DSBD_AUDIT_FAQUAL	(457,8)	

Note: In the example to follow, both a COPY and SORT statement are used to show how the output from one operator can be used as input to another; this particular set of tasks could have been done with one SORT operator

ICETOOL Problem Example

```
//STEPICE  EXEC PGM=ICETOOL,REGION=100M
…
//DBUNLD    DD DSN=RLW.R16DBUA,DISP=SHR
//LVLRPT1   DD DSN=RLW.SORTED1,DISP=SHR
//LVLRPT2   DD DSN=RLW.SORTED2,DISP=SHR
//LISTOUT   DD SYSOUT=*
//SYSOUT    DD SYSOUT=*
//TOOLIN    DD *
COPY FROM(DBUNLD) TO(LVLRPT1) USING(LVL1)
SORT FROM(LVLRPT1) TO(LVLRPT2) USING(LVL2)
DISPLAY FROM(LVLRPT2) LIST(LISTOUT) -
HEADER('PROFILE')        ON(10,44,CH) -
HEADER('LEVEL')          ON(175,2,CH) -
HEADER('AUDIT LEVEL')    ON(147,8,CH) -
HEADER('AUDIT SUCCESS')  ON(452,8,CH) -
HEADER('AUDIT FAILURES') ON(461,8,CH)
//LVL1CNTL  DD *
 OPTION VLSHRT
 OUTFIL FNAMES=LVLRPT1,
 INCLUDE=((5,4,CH,EQ,C'0400'),AND,(174,3,CH,NE,C'000'))
//LVL2CNTL  DD *
 SORT FIELDS=(174,3,CH,A)
```

Hints and Tips

ICEGENER

- DFSORT program replacement for z/OS utility IEBGENER
- Uses EXCP processing for I/O
- Performance improvements depend on file size
- Can realize from 50% to 90% reduction in CPU utilization

Hints and Tips

SYMBOLIC NAMES

- Can be used to define fields and constants
- Similar to Copy Books and DSECTs
- Provide symbolic definitions in DD statement SYMNAMES
- DFSORT shows substitutions in DD statement SYMNOUT
- Addresses DFSORT constraints
- User needs to do the research and preparation of input.

Example:

```
//SYMNAMES DD *
RECTYPE,1,4,CH
GROUPTYPE,C'0101'
OWNER-GROUP,10,8,CH
//DFSPARM  DD *
 INCLUDE=(RECTYPE,EQ,GROUPTYPE,AND,GROUPNAME,EQ,C'SYS1    ')
```

Hints and Tips

PREPARE AND TEST

- Experiment with operators and operands
- Become familiar with files you have available for use
- Use DFSORT to assist programs that filter input
- Don’t wait until there is a deadline
  - Prepare record layouts for files
  - Build dataset and store Symbolic Names
  - Build JCL library with a example of each control statement
  - Test each member in the JCL library

References

IBM Manuals

- SC26-7524 z/OS DFSORT Installation and Customization
- SC26-7527 z/OS DFSORT: Getting Started
- SC26-7523 z/OS DFSORT: Application Programming Guide
- SC26-7525 z/OS DFSORT: Messages, Codes, and Diagnosis Guide
- SC26-7526 z/OS DFSORT: z/OS DFSORT Tuning Guide
- GC26-7037 DFSORT Panels Guide
- SA22-7682 Security Server RACF Macros and Interfaces
- SA22-7684 Security Server RACF Auditor's Guide
- SG24-5627 OS/390 Security Server 1999 Updates Technical Presentation Guide

DFSORT Home Page

- http://ibm.com/storage/dfsort

Ask Professor Sort

- http://ibm.com/systems/support/storage/software/sort/mvs/professor_sort

"RACF and DFSORT Security Analysis Tools", Mark Nelson & Frank Yaeger, Technical Support, October 1996

References

IBM Supplied ICETOOL Examples

- SYS1.SAMPLIB(IRRICE)

RACFICE Downloads Page

- http://ibm.com/servers/eserver/zseries/zos/racf/racfice.html
- The RACFICE tool consists of these files:
  - Documentation for RACFICE in PDF format (105K)
  - Source code for RACFICE in TSO TRANSMIT format (652K)
  - Sample IRRDBU00 data in TSO TRANSMIT format (625K)
  - Sample IRRADU00 data in TSO TRANSMIT format (6.2M)
  - Source code for RACFICE in IEBUPDTE format (653K)
- Files include SYMNAMES for the unload file record fields
- Since the introduction of the SYS1.SAMPLIB member with OS/390 2.8, these files have not been maintained