

# DFSORT: ICETOOL Mini-User Guide

December, 2004

Frank L. Yaeger

DFSORT Team  
IBM Systems Software Development  
San Jose, California  
Internet: [yaeger@us.ibm.com](mailto:yaeger@us.ibm.com)

## DFSORT Web Site

For papers, online books, news, tips, examples and more, visit the DFSORT home page at URL:

<http://www.ibm.com/storage/dfsor>



---

## **Abstract**

This paper is a mini-user guide for DFSORT's versatile ICETOOL data processing and reporting utility. The major features of ICETOOL for z/OS DFSORT V1R5 and DFSORT Release 14, including its JCL and control statements, are discussed at length using many examples. The objective is to show you how to use DFSORT's ICETOOL to accomplish complex tasks.



---

# Contents

<b>ICETOOL Mini-User Guide</b>	<b>1</b>
Introduction	1
Additional Sources for ICETOOL Examples	1
What Can ICETOOL Do?	2
General ICETOOL Job Format	4
Required JCL Statements	4
ICETOOL Operator Statement Syntax	4
Operator Statements	4
Comment Statements	5
Blank Statements	5
ICETOOL Return Codes	5
ICETOOL Job with all Operators	6
COPY Operator Details	9
Syntax	9
Function	9
Examples	9
Required Operands	9
Optional Operands	10
Note	10
Example of TOOLMSG Output for COPY	10
COUNT Operator Details	11
Syntax	11
Function	11
Examples	11
Required Operand	11
Optional Operands	12
Symbols	12
Notes	12
Example of TOOLMSG Output for COUNT	12
Example of Using COUNT to Set RC=12 or RC=0	12
Example of Using COUNT to Set RC=4 or RC=0	12
DEFAULTS Operator Details	14
Syntax	14
Function	14
Examples	14
Required Operand	14
Optional Operands	14
Notes	15
Example of TOOLMSG Output for DEFAULTS	15
Example of List Output for DEFAULTS	15
DISPLAY Operator Details	17
Syntax	17
Function	17
Examples	18
Required Operands	18
Optional Operands	21
Symbols	22
Notes	22
Example of TOOLMSG Output for DISPLAY	23
Example of List Output for DISPLAY	23

Example using OUTFIL Lookup and Change with DISPLAY	24
Example of Lookup and Change with DISPLAY Output	24
Example of Plain and Fancy Reports	25
Example of Plain Report Output	26
Example of Fancy Report Output	26
MODE Operator Details	27
Syntax	27
Function	27
Examples	27
Required Operand	27
Note	27
Example of TOOLMSG Output for MODE	27
OCCUR Operator Details	28
Syntax	28
Function	28
Examples	29
Required Operands	29
Optional Operands	31
Symbols	32
Notes	32
Example of TOOLMSG Output for OCCUR	33
Example of List Output for OCCUR	33
Example of HIGHER with OCCUR	33
Example of HIGHER with OCCUR Output	33
RANGE Operator Details	34
Syntax	34
Function	34
Examples	34
Required Operands	34
Optional Operand	35
Symbols	35
Note	35
Example of TOOLMSG Output for RANGE	35
SELECT Operator Details	36
Syntax	36
Function	36
Examples	36
Required Operands	36
Optional Operands	37
Symbols	37
Notes	38
Example of TOOLMSG Output for SELECT	39
SORT Operator Details	40
Syntax	40
Function	40
Examples	40
Required Operands	41
Optional Operands	41
Notes	41
Example of TOOLMSG Output for SORT	41
SPLICE Operator Details	42
Syntax	42
Function	42
Examples	42

Required Operands	43
Optional Operands	43
Symbols	44
Notes	44
Example of TOOLMSG Output for SPLICE	45
STATS Operator Details	46
Syntax	46
Function	46
Examples	46
Required Operands	46
Optional Operand	46
Symbols	47
Note	47
Example of TOOLMSG Output for STATS	47
UNIQUE Operator Details	48
Syntax	48
Function	48
Examples	48
Required Operands	48
Optional Operands	48
Symbols	48
Notes	49
Example of TOOLMSG Output for UNIQUE	49
VERIFY Operator Details	50
Syntax	50
Function	50
Examples	50
Required Operands	50
Optional Operands	50
Symbols	50
Notes	51
Example of TOOLMSG Output for VERIFY	51
Symbol Processing	52
Calling ICETOOL from a Program	53
<b>Appendix A. Edit Masks for DISPLAY and OCCUR Formatting</b>	<b>55</b>





---

# ICETOOL Mini-User Guide

---

## Introduction

ICETOOL, a versatile data set processing and reporting utility, provides an easy-to-use batch front-end for DFSORT. ICETOOL combines new features with previously available DFSORT features to perform complex sorting, copying, reporting and analytical tasks using multiple data sets in a single job step. ICETOOL was first introduced in DFSORT Release 11.1 and was significantly enhanced in Releases 12, 13 and 14. DFSORT R14 PTF UQ99331 (March, 2002), DFSORT R14 PTF UQ90053 (February, 2003), DFSORT R14 PTF UQ95213 (December, 2004), and z/OS DFSORT V1R5 PTF UQ95214 (December, 2004) provided many new significant enhancements to ICETOOL which are reflected in this paper.

This paper is a mini-user guide for ICETOOL. The major features of ICETOOL for z/OS DFSORT V1R5 and DFSORT Release 14, including its JCL and control statements, are discussed at length using many examples. The objective is to show you how to use DFSORT's ICETOOL to accomplish complex tasks.

Complete information on ICETOOL's JCL, control statements, restrictions, and calling program interface can be found in *DFSORT Application Programming Guide (SC26-7523)*. "Examples of DFSORT Job Streams" in *DFSORT Application Programming Guide* contains a sample ICETOOL job that shows how ICETOOL can be used to perform complex tasks using multiple operations and data sets in a single step, along with a complete explanation of the job.

Complete information on ICETOOL's messages and return codes can be found in *DFSORT Messages, Codes and Diagnosis Guide (SC26-7525)*.

Complete information on the newest features of ICETOOL (UFF, SFF, DCn, DEn, TCn, TEn, larger numbers and constants, multiline headings, Gn, YDDD, YDDDNS, RC4, KEEPBASE, VLENMAX and VLENOVLY) can be found in the z/OS DFSORT V1R6 books, and in the *User Guide for DFSORT PTFs UQ95214 and UQ95213* paper.

You can access all of these documents online by clicking the **Publications** link on the DFSORT home page at URL:

<http://www.ibm.com/storage/dfsort>

---

## Additional Sources for ICETOOL Examples

- The DFSORT home page on the World Wide Web at URL:  
<http://www.ibm.com/storage/dfsort>
- *DFSORT Application Programming Guide (SC26-7523)*
- *DFSORT: Getting Started (SC26-7527)*
- *z/OS DFSMSrmm Reporting (SC26-7406)*
- *z/OS DFSMSshm 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)*
- RACFICE describes a technique for analyzing RACF data using ICETOOL. You can obtain RACFICE at:  
<http://www.ibm.com/servers/eserver/zseries/zos/racf/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 Example 1: VSAM report
  - DCOLEX2 - DCOLLECT Example 2: Conversion reports
  - 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/dfsor/mvs/>

---

## What Can ICETOOL Do?

ICETOOL is a versatile DFSORT utility that allows you to perform multiple operations on one or more data sets in a single job step.

ICETOOL uses the capabilities of DFSORT to perform the operations you request, calling DFSORT for each operation with the particular DFSORT control statements and options required.

The thirteen ICETOOL operators, each of which can be used one or more times in a single run, allow you to perform a variety of functions such as:

- Displaying statistical information for selected numeric fields, such as minimum, maximum, average, total, count of values within a range and count of unique values. This makes it easy to extract frequently used analytical data.
- Displaying list data sets showing character (up to 1500 bytes) and numeric fields in a variety of report formats, allowing control of title, date, time, page numbers, headings, lines per page, field formats, and total, maximum, minimum and average values for the columns of numeric data. This makes it easy to create simple, tailored and sectioned reports.
- Identifying and displaying invalid decimal values and their locations in a data set. This makes it easy to avoid using invalid fields for other operations or steps.
- Creating multiple copies of sorted, edited, or unedited data sets. This makes it easy to create several identical data set copies.
- Creating output data sets containing different subsets or field arrangements of input data sets. This makes it easy to view data in many different ways.
- Displaying a list data set showing the DFSORT installation defaults selected at your site. This makes it easy to determine the options selected and accepted for all eight installation environments.
- Creating output data sets and displaying list data sets for records with duplicate values, non-duplicate values, or values that occur n times, less than n times, or more than n times. This makes it easy to view data according to occurrences of values.
- Creating list data sets showing unique values for selected character and numeric fields and the number of times each occurs, in a variety of report formats. This makes it easy to obtain reports based on occurrences of values.

- Creating output data sets with information spliced together from two or more input records with duplicate values. The information in the input records can originate from different data sets. This makes it easy to perform various file "join" and "match" operations.
- Displaying the count of records in a data set. This makes it easy to determine the number of records in a data set or in a subset of a data set.
- Setting RC=12, RC=4 or RC=0 based on the count of records in a data set. This makes it easy to avoid using data sets that are empty, not empty or contain a certain number of records, for other operations or steps.
- Allowing operations to be performed or suppressed based on the success or failure of previous operations. This makes it easy to group operations according to the action to be taken after an error.

ICETOOL can be called directly or from a program. Operators can be supplied in a data set. Alternatively, operators can be supplied by a calling program parameter list, in which case ICETOOL returns information for each operation in the parameter list. In either case, ICETOOL prints messages and gives a return code for each operation.

The thirteen ICETOOL operators are: COPY, COUNT, DEFAULTS, DISPLAY, MODE, OCCUR, RANGE, SELECT, SORT, SPLICE, STATS, UNIQUE, and VERIFY. By using combinations of these thirteen operators, you can easily create applications that perform many complex tasks. By supplying these operators in a parameter list, you can use the information returned by ICETOOL in your program.

---

## General ICETOOL Job Format

Here's a representation of an ICETOOL job:

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

### Required JCL Statements

- 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 Operator Statement Syntax

### Operator Statements

- Each ICETOOL operator statement describes a task you want ICETOOL to perform.
- Any number of operators can be specified and in any order.
- The general format for all ICETOOL statements is:

**operator operand ... operand**

Example:

```
COPY FROM(IN) TO(OUT1,OUT2)
```

- **operator** is one of the thirteen 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)
```

## Comment Statements

- Asterisk (\*) in column 1 indicates a comment statement.
- Comment statements are printed with other ICETOOL statements, but otherwise ignored.

## Blank Statements

- Blank in columns 1-72 indicates a blank statement.
- Blank statements are ignored since ICETOOL prints blank lines where appropriate.

---

## ICETOOL Return Codes

ICETOOL sets a return code for each operation it performs.

For the step, ICETOOL sets the return code to the highest operator return code.

The return codes are:

- 0 - Successful completion. No errors were detected.
- 4 - Successful completion. DFSORT detected one or more warning conditions.
- 12 - Unsuccessful completion. ICETOOL detected one or more errors. Can also be set if the record count meets a specified criteria (for example, a data set is empty, or a data set contains more than 50000 records).
- 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. The operating system was not z/OS or OS/390.

---

## ICETOOL Job with all Operators

Here's a sample ICETOOL job that briefly explains the function of each ICETOOL operator and shows a simple example of its use. Later pages explain each operator in the job and its parameters (required and optional), and show the TOOLMSG messages produced for each operator.

```
//EXAMP JOB ...
//* JCL REQUIRED FOR ALL ICETOOL JOBS
//TOOL EXEC PGM=ICETOOL,REGION=1024K
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//TOOLIN DD *
* ICETOOL Operators - brief explanations and simple examples

* COPY - copies a data set to one or more output data sets.
* Multiple output is handled using a single pass over
* the input.
* Example: copy the IN data set to the DASD, PRINT and TAPE
* data sets.
  COPY FROM(IN) TO(DASD,PRINT,TAPE)

* COUNT - prints a message in TOOLMSG containing the count of
* records in a data set. Can also be used to set RC=12 or RC=0
* based on the count of records in a data set.
* Example: print a count of the number of records in the
* IN2 data set.
  COUNT FROM(IN2)

* DEFAULTS - prints the DFSORT installation defaults in a
* separate list data set.
* Example: print the JCL, INV, TSO, TSOINV, TD1, TD2,
* TD3 and TD4 installation defaults selected, in the
* INSTDEF data set
  DEFAULTS LIST(INSTDEF)

* 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.
* Example: print a tailored report showing values from the
* three IN2 data set ON fields, with title, heading lines,
* maximum and minimum, in the DOUT data set.
  DISPLAY FROM(IN2) LIST(DOUT) BLANK -
  TITLE('IN2 Data Set Report') PAGE DATE TIME -
  HEADER('Store') ON(50,15,CH) -
  HEADER('Profit/(Loss)') ON(26,8,ZD,E1) -
  HEADER('Employees') ON(2,3,ZD) -
  MAXIMUM('Largest') MINIMUM('Smallest')
```

- \* MODE - specifies the error checking and actions after error detection to be performed for a group of operators.
- \* Example: continue processing operators whether or not an error is detected.

```
MODE CONTINUE
```

- \* OCCUR - prints each unique value for specified numeric and 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 meets specified criteria (e.g. only duplicate values).
- \* Example: print a report showing values from the IN1 data set ON field, with the number of times each value occurs, in the LIST1 data set.

```
OCCUR FROM(IN1) LIST(LIST1) TITLE('Dep't Counts') -
  ON(35,3,CH) ON(VALCNT)
```

- \* RANGE - prints a message in TOOLMSG containing the count of values in a specified range for a specified numeric field in a data set.
- \* Example: print a count of the values in the IN2 data set ON field that are higher than -50, but lower than +100.

```
RANGE FROM(IN2) ON(20,2,PD) HIGHER(-50) LOWER(100)
```

- \* 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 (e.g. only duplicate values).
- \* Records that are not selected can be saved in a separate output data set.
- \* Example: selects records from the IN1 data set, for the SEL1 data set, whose ON field occurs only once (i.e., only records with no duplicate ON field values).

```
SELECT FROM(IN1) TO(SEL1) ON(30,2,PD) NODUPS
```

- \* SORT - sorts a data set to one or more output data sets. Multiple output is handled using a single pass over the input.
- \* Example: sort the IN1 data set to the OUT1 data set using the DFSORT control statements in the CTL1CNTL data set.

```
SORT FROM(IN1) TO(OUT1) USING(CTL1)
```

- \* SPLICE - splices together specified fields from records that have the same specified numeric or character field values (i.e., 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 various "join" and "match" operations can be performed.
- \* Example: for ON fields that occur more than once (i.e., duplicate ON field values) in the CONCT data set, splices the WITH field from the last duplicate record into the first duplicate record.

```
SPLICE FROM(CONCT) TO(COMBINE) ON(11,5,CH) WITH(41,20)
```

```

* STATS - prints messages in TOOLMSG containing the minimum,
* maximum, average, and total for specified numeric fields in
* a data set.
* Example: print the minimum, maximum, average and total
* values for the three VLRIN data set ON fields.
* For variable-length records, ON(VLEN) gives statistics
* about the length of the records.
  STATS FROM(VLRIN) ON(VLEN) ON(12,2,ZD) ON(18,5,FS)

* UNIQUE - prints a message in TOOLMSG containing the count of
* unique values for a specified numeric or character field.
* Example: print the count of unique values in the
* OUT1 data set ON field.
  UNIQUE FROM(OUT1) ON(30,2,PD)

* VERIFY - examines specified decimal fields in a data set and
* prints a message in TOOLMSG identifying each invalid value
* found for each field.
* Example: identify all values in the two IN2 data set
* decimal ON fields that have invalid digits (A-F)
* and/or invalid signs (0-9).
  VERIFY FROM(IN2) ON(10,2,ZD) ON(41,6,PD)
/*
/** JCL REQUIRED FOR THE SPECIFIED OPERATORS
//N DD DSN=Y897797.NAMES,DISP=SHR
//DASD DD DSN=Y897797.OUT.COPY,DISP=OLD
//PRINT DD SYSOUT=A
//TAPE DD DSN=TAPE1,UNIT=3490,DISP=(,KEEP),VOL=SER=VOL001,
// LABEL=(,SL)
//INSTDEF DD SYSOUT=A
//IN2 DD DSN=Y897797.FIXED2.IN,DISP=SHR
//DOUT DD SYSOUT=A
//IN1 DD DSN=Y897797.FIXED1.IN,DISP=SHR
//LIST1 DD SYSOUT=A
//SEL1 DD DSN=&&SEL1,DISP=(,PASS),SPACE=(TRK,(5,5)),UNIT=SYSDA
//OUT1 DD DSN=&&TEMP,DISP=(,PASS),SPACE=(TRK,(5,5)),UNIT=SYSDA
//VLRIN DD DSN=Y897797.VARIABLE.IN,DISP=SHR
//CONCT DD DSN=Y897797.BASE.IN,DISP=SHR
//      DD DSN=Y897797.OVERLAY.IN,DISP=SHR
//COMBINE DD DSN=Y897797.COMBINE.OUT,DISP=(NEW,CATLG,DELETE),
// SPACE=(CYL,(5,5)),UNIT=SYSDA
//CTL1CNTL DD *
  SORT FIELDS=(35,3,CH,A)
  OMIT COND=(35,3,SS,EQ,C'J82,D54')
/*

```



---

## COPY Operator Details

### Syntax

```
COPY FROM(indd) TO(outdd,...) USING(yyyy) VSAMTYPE(x)
```

```
    LOCALE(name)    SERIAL
    LOCALE(CURRENT)
    LOCALE(NONE)
```

### Function

Copies the indd data set to the outdd data sets (up to 10). The DFSORT control statements in yyyyCNTL are used if USING(yyyy) is specified. DFSORT control statements and options can be used to copy a subset of the input records (INCLUDE or OMIT statement; SKIPREC and STOPAFT options; OUTFIL INCLUDE, OMIT, STARTREC, ENDREC and SPLIT operands), reformat records for output (INREC, OUTREC and OUTFIL), and so on. See *DFSORT Application Programming Guide* for complete details of DFSORT control statements and options.

### Examples

```
COPY FROM(IN1) TO(NEW,BACKUP) USING(CTL1)
```

```
COPY FROM(VSAMIN) TO(VSAMOUT) VSAMTYPE(V)
```

```
COPY FROM(MASTER) USING(OUTF)
```

CTL1CNTL might contain the following:

```
//CTL1CNTL DD *
  OMIT COND=(28,3,SS,EQ,C'J69,L92,J82')
/*
```

Note the use of DFSORT's substring search feature to OMIT the records for 'J69', 'L92' and 'J82' with a single condition. Complete details on this feature of INCLUDE/OMIT as well as its many other features can be found in *DFSORT Application Programming Guide*.

OUTFCNTL might contain the following:

```
//OUTFCNTL DD *
  OUTFIL FNames=FMT1,OUTREC=(C'SUB1',32,15,2Z,1,3,PD)
  OUTFIL FNames=FMT2,OUTREC=(C'SUB2',16,2,BI,M1,C'**,32,15)
/*
```

Note that DFSORT's OUTFIL OUTREC feature is used here to create two output data sets with different field arrangements using a single pass over the input data set. Complete details on this feature of OUTFIL as well as its many other features can be found in *DFSORT Application Programming Guide*.

### Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- TO - the ddnames of 1 to 10 output data sets. You must supply DD statements for the ddnames you specify. TO, USING, or TO and USING must be specified.
- USING - the first 4 characters of the ddname (yyyyCNTL) for the DFSORT control statement data set. You

must supply a DD statement for xxxxCNTL if you specify USING(yyyy). TO, USING, or TO and USING must be specified.

## Optional Operands

- VSAMTYPE - the record format for a VSAM input data set (F or V).
- LOCALE - overrides the installation default for locale processing.
- SERIAL - causes OUTFIL processing not to be used for multiple outdd data sets. SERIAL is not recommended because it imposes data set restrictions and can degrade performance.

## Note

- If you use OUTFIL statements in xxxxCNTL to specify your output data sets, you do not need to specify TO.

## Example of TOOLMSG Output for COPY

```
* COPY - copies a data set to one or more output data sets.
* Multiple output is handled using a single pass over
* the input.
* Example: copy the N data set to the DASD, PRINT and TAPE
* data sets.
  COPY FROM(N) TO(DASD,PRINT,TAPE)
ICE627I 0 DFSORT CALL 0001 FOR COPY FROM N          TO OUTFIL  COMPLETED
ICE602I 0 OPERATION RETURN CODE:  00
```

Note that DFSORT's OUTFIL multiple output feature is used automatically to create the three output data sets with a single pass over the input data set.

---

## COUNT Operator Details

### Syntax

```
COUNT FROM(indd) USING(xxxx) VSAMTYPE(x) LOCALE(name)
                                           LOCALE(CURRENT)
                                           LOCALE(NONE)

      EMPTY          RC4
      NOTEMPTY
      HIGHER(x)
      LOWER(y)
      EQUAL(v)
      NOTEQUAL(w)
```

### Function

Prints a message in TOOLMSG containing the count of records in the indd data set. The DFSORT control statements in xxxxCNTL are used if USING(xxxx) is specified. The INCLUDE or OMIT statement can be used to count a subset of the input records.

COUNT can also be used to set RC=12 or RC=4 if the record count meets specified criteria (that is, empty, not empty, n records, not n records, more than n records or less than n records), or RC=0 if the record count does not meet specified criteria.

### Examples

```
COUNT FROM(IN1)

COUNT FROM(INPUT5) EMPTY

COUNT FROM(MASTER) HIGHER(50000) RC4

COUNT FROM(IN1) USING(BITL)

COUNT FROM(UPDATE2) USING(BITL) NOTEMPTY
```

BITLCNTL might contain the following:

```
//BITLCNTL DD *
  INCLUDE COND=(15,1,BI,EQ,B'10.....1')
/*
```

Note the use of DFSORT's bit logic feature to INCLUDE records with bit 0 on, bit 1 off and bit 7 on in byte 15. Complete details on this feature of INCLUDE/OMIT as well as its many other features can be found in *DFSORT Application Programming Guide*.

### Required Operand

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.

## Optional Operands

- USING - the first 4 characters of the ddname (xxxxCNTL) for the DFSORT control statement data set. You must supply a DD statement for xxxxCNTL if you specify USING(xxxx).
- VSAMTYPE - the record format for a VSAM input data set (F or V).
- LOCALE - overrides the installation default for locale processing.
- EMPTY, NOTEMPTY, HIGHER, LOWER, EQUAL, NOTEQUAL - defines the criteria against which the record count is to be matched. If the criteria is met, ICETOOL sets RC=12 for this COUNT operator by default, or RC=4 if RC4 is specified. If the criteria is not met, ICETOOL sets RC=0 for this COUNT operator. x, y, v, and w must be specified as n or +n where n can be 0 to 562949953421310.
- RC4 - sets RC=4 if the criteria is met (overriding the default of RC=12).

## Symbols

- Symbols for decimal constants can be used instead of n and +n in the HIGHER, LOWER, EQUAL and NOTEQUAL operands.

## Notes

- The record count is also printed for the DISPLAY, OCCUR, RANGE, SELECT, STATS, UNIQUE, and VERIFY operators.
- You can use COUNT to set RC=12 based on the record count to stop other operations or steps from executing. When you specify one of the criteria operands, ICETOOL does not print the record count, and it uses DFSORT's STOPAFT option to process the minimum number of records required to determine whether or not the criteria is met.

## Example of TOOLMSG Output for COUNT

```
* COUNT - prints a message in TOOLMSG containing the count of
* records in a data set.
* Example: print a count of the number of records in the
* IN2 data set.
COUNT FROM(IN2)
ICE627I 0 DFSORT CALL 0002 FOR COPY FROM IN2      TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 0000000000000008
ICE602I 0 OPERATION RETURN CODE: 00
```

## Example of Using COUNT to Set RC=12 or RC=0

```
COUNT FROM(IN5) EMPTY
```

### TOOLMSG Output if IN5 is empty

```
ICE646A 0 RECORD COUNT MEETS CRITERIA - RC=12 SET
ICE602I 0 OPERATION RETURN CODE: 12
```

### TOOLMSG Output if IN5 is not empty

```
ICE647I 0 RECORD COUNT DOES NOT MEET CRITERIA - RC=0 SET
ICE602I 0 OPERATION RETURN CODE: 00
```

## Example of Using COUNT to Set RC=4 or RC=0

```
COUNT FROM(IN5) HIGHER(5) RC4
```

**TOOLMSG Output if IN5 has 6 or more records**

ICE651A 0 RECORD COUNT MEETS CRITERIA - RC=4 SET  
ICE602I 0 OPERATION RETURN CODE: 04

**TOOLMSG Output if IN5 has 5 or less records**

ICE647I 0 RECORD COUNT DOES NOT MEET CRITERIA - RC=0 SET  
ICE602I 0 OPERATION RETURN CODE: 00

---

# DEFAULTS Operator Details

## Syntax

```
DEFAULTS LIST(listdd) LISTSDB  
                    LISTNOSDB
```

## Function

Prints, in the listdd data set, the DFSORT installation defaults. DFSORT lets you maintain eight separate sets of installation defaults using eight installation modules as follows:

- Environment installation modules
  - JCL (ICEAM1 module) - batch JCL directly invoked installation module
  - INV (ICEAM2 module) - batch program invoked installation module
  - TSO (ICEAM3 module) - TSO directly invoked installation module
  - TSOINV (ICEAM4 module) - TSO program invoked installation module
- Time-of-day installation modules
  - TD1 (ICETD1 module) - first time-of-day installation module
  - TD2 (ICETD2 module) - second time-of-day installation module
  - TD3 (ICETD3 module) - third time-of-day installation module
  - TD4 (ICETD4 module) - fourth time-of-day installation module

DEFAULTS produces a report showing the installation defaults for ICEAM1-4 followed by the installation defaults for ICETD1-4. The value for each item (for each of the eight installation modules) is shown as it is set in the ICEAM1-4 and ICETD1-4 modules loaded from the STEPLIB, JOBLIB or link library. For any value that is different from the IBM-supplied value, the IBM-supplied value is shown below it.

See *DFSORT Installation and Customization* for complete details of the eight installation modules and the installation defaults and their values.

## Examples

```
DEFAULTS LIST(SHOWDEF)
```

```
DEFAULTS LIST(DFLT) LISTSDB
```

## Required Operand

- LIST - the ddname of the list data set. You must supply a DD statement for the ddname you specify.

## Optional Operands

- LISTSDB - uses SDB for the list data set (overrides installation value SDBMSG=NO).
- LISTNOSDB - does not use SDB for the list data set (overrides installation value SDBMSG=YES).

## Notes

- The control character occupies the first byte of each record in the list data set.
- RECFM=FBA and LRECL=121 are used for the list data set.

## Example of TOOLMSG Output for DEFAULTS

```
* DEFAULTS - prints the DFSORT installation defaults in a
* separate list data set.
* Example: print the JCL, INV, TSO, TSOINV, TD1, TD2,
* TD3 and TD4 installation defaults selected, in the
* INSTDEF data set
  DEFAULTS LIST(INSTDEF)
ICE603I 0 INFORMATION PRINTED IN INSTDEF  DATA SET
ICE602I 0 OPERATION RETURN CODE:  00
```

## Example of List Output for DEFAULTS

Here's an example of the output that would appear in INSTDEF:

Z/OS DFSORT V1R5 INSTALLATION (ICEAMC) DEFAULTS - 1 -

\* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE  
SPECIFIED DEFAULT)

ITEM	JCL (ICEAM1)	INV (ICEAM2)	...
-----	-----	-----	...
RELEASE	V1R5	V1R5	...
MODULE	ICEAM1	ICEAM2	...
APAR LEVEL	BASE	BASE	...
COMPILED	07/21/03	07/21/03	...
ENABLE	NONE	TD1	...
ABCODE	MSG	99	...
		* MSG	...
ALTSEQ	SEE BELOW	SEE BELOW	...
ARESALL	0	0	...
ARESINV	NOT APPLICABLE	0	...
CFW	YES	YES	...
CHALT	YES	YES	...
	* NO	* NO	...
.	.	.	...
.	.	.	...
.	.	.	...

\* IBM-SUPPLIED DEFAULT (ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED DEFAULT)

ITEM	TD1 (ICETD1)	TD2 (ICETD2)	...
-----	-----	-----	...
RELEASE	V1R5	V1R5	...
MODULE	ICETD1	ICETD2	...
APAR LEVEL	BASE	BASE	...
COMPILED	07/21/03	07/21/03	...
SUN	0600-2000	NONE	...
	* NONE		
MON	NONE	NONE	...
TUE	NONE	NONE	...
WED	NONE	NONE	...
THU	NONE	NONE	...
FRI	NONE	NONE	...
SAT	0600-2000	NONE	...
	* NONE		
ABCODE	99	MSG	...
	* MSG		...
ALTSEQ	SEE BELOW	SEE BELOW	...
ARESALL	0	0	...
ARESINV	0	0	...
CFW	YES	YES	...
CHALT	YES	NO	...
	* NO		...
.	.	.	...
.	.	.	...
.	.	.	...



---

## DISPLAY Operator Details

### Syntax

```
DISPLAY FROM(indd) LIST(listdd) ON(p,m,f)          ...
                                ON(p,m,f,formatting)
                                ON(p,m,HEX)
                                ON(VLEN)
                                ON(VLEN,formatting)
                                ON(NUM)
                                ON(NUM,formatting)

                                TITLE('string') PAGE DATE          TIME
                                                DATE(abcd) TIME(abc)
                                                DATENS(abc) TIMENS(ab)
                                                YDDD(abc)
                                                YDDDNS(ab)

                                HEADER('string1')          ... LINES(n) BLANK
                                HEADER('string1','string2') PLUS
                                HEADER('string1','string2','string3')
                                HEADER(NONE)
                                NOHEADER

                                TOTAL('string') MAXIMUM('string') MINIMUM('string')

                                AVERAGE('string') LIMIT(n) VSAMTYPE(x) WIDTH(n)

                                BREAK(p,m,f)          BTITLE('string') BTOTAL('string')
                                BREAK(p,m,f,formatting)

                                BMAXIMUM('string') BMINIMUM('string') BAVERAGE('string')

                                INDENT(n) BETWEEN(n) STATLEFT UZERO LISTSDB
                                                LISTNOSDB
```

### Function

Prints, in the listdd data set, the values or characters of specified numeric (including SMF, TOD and ETOD date/time) and/or character fields. The fields are printed in columns in the same order in which they are specified. From 1 to 20 ON fields can be specified as long as the resulting list data set line length does not exceed the limit specified by the WIDTH operand or 2048 bytes.

Simple, tailored or sectioned reports can be produced:

- A title line consisting of a string, the page number, the date and/or the time, in any order specified, can appear at the top of each page.
- One, two or three line headings can be printed to identify each specified field.
- Overall statistics (total, maximum, minimum, and/or average) can be produced for numeric fields.
- Column widths are adjusted automatically according to the operands specified and the length of the fields.
- Alternate formats for date and time can be specified.

- Numeric fields, consisting of a sign and up to 31 digits in various formats, can be edited with separators, decimal points, patterns, division, and leading, trailing and floating signs.
- A break field can be used to produce sections with their own break title and break statistics (total, maximum, minimum and/or average).

## Examples

```
DISPLAY FROM(DATA) LIST(SHOWIT) ON(10,44,CH) ON(5,4,FS)
```

```
DISPLAY FROM(VLR) LIST(RDWLIST) -
  INDENT(10) BETWEEN(5) -
  HEADER('Record','number') ON(NUM,N04) -
  HEADER('Record','length') ON(VLEN,N05) -
  HEADER('RDW in Hex') ON(1,4,HEX) -
  TITLE('Record Length Report') DATE -
  TOTAL('Total') AVERAGE('Average') -
  MINIMUM('Minimum') MAXIMUM('Maximum')
```

```
DISPLAY FROM(MASTER) LIST(DENMARK) LINES(50) -
  PAGE TITLE('Report for Denmark') DATE(DM4-) -
  BTITLE('Division:') BREAK(23,10,CH) -
  HEADER('Part') ON(15,6,CH) -
  HEADER('Completed') ON(3,4,ZD,A2) -
  HEADER('Value (kr)') ON(38,12,SFF,C2) -
  BMINIMUM('Lowest in this Division:') -
  BMAXIMUM('Highest in this Division:') -
  BAVERAGE('Average in this Division:') -
  BTOTAL('Total for this Division:') -
  AVERAGE('Average for all Divisions:')
```

```
DISPLAY FROM(DATA1) LIST(KBDATA) WIDTH(150) -
  NOHEADER ON(15,44,CH) ON(8,4,PD,/KB,T' KB')
```

```
DISPLAY FROM(SMF14) LIST(SMF14RPT) -
  TITLE('SMF Type-14 Records') DATENS(4MD) -
  LISTSDB -
  HEADER('Date') ON(11,4,DT1,E'9999/99/99') -
  HEADER('Time') ON(7,4,TM1,E'99:99:99') -
  HEADER('Sys') ON(15,4,CH) -
  HEADER('Jobname') ON(19,8,CH) -
  HEADER('Datasetname') ON(69,44,CH)
```

## Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- LIST - the ddname of the list data set. You must supply a DD statement for the ddname you specify.
- ON - a field to be used for this operation. From 1 to 20 ON fields can be specified.
  - (p,m,f) gives the position, length and format of a numeric or character field. A field must not extend beyond position 32752 or the end of the record.
  - (p,m,f,formatting) gives the position, length and format of a numeric or character field and specifies how the data for this field is to be formatted for printing. A field must not extend beyond position 32752 or the end of the record.

The following formatting items can be used (see *DFSORT Application Programming Guide* for complete details):

- Mask - an edit mask to be applied to the numeric data. Thirty-nine pre-defined edit masks are available, encompassing many of the numeric notations throughout the world with respect to separators, decimal point, decimal places, signs and so on. See "Appendix A" for complete descriptions and examples of all thirty-nine masks. The attributes of each group of masks is shown below.

<b>Masks</b>	<b>Separators</b>	<b>Decimal Places</b>	<b>Positive Sign</b>	<b>Negative Sign</b>
A0	No	0	blank	-
A1-A5	Yes	0	blank	-
B1-B6	Yes	1	blank	-
C1-C6	Yes	2	blank	-
D1-D6	Yes	3	blank	-
E1-E4	Yes	0	blank	( )
F1-F5	Yes	2	blank	( )
G1-G6	Yes	4	blank	-

- E'pattern' - an edit pattern (1 to 44 characters) to be applied to the numeric data. Each **9** in the pattern (up to 31) is replaced by a corresponding digit (0-9) from the numeric value. Other characters in the pattern appear as specified. E'pattern' can be used for formatting unsigned numeric data such as telephone numbers, dates, and so on.
- L'string' - a leading string to appear at the beginning of the character or numeric data column. Each string can be 1 to 10 characters.
- F'string' - a floating string to appear to the left of numeric data. Each string can be 1 to 10 characters.
- T'string' - a trailing string to appear at the end of the character or numeric data column. Each string can be 1 to 10 characters.
- LZ - used with an edit mask to print leading zeros for the numeric field (overriding the default of suppressing leading zeros).
- NOST - used to suppress specified statistics for the numeric field (overriding the default of printing specified statistics).
- Ndd - use dd digits for the numeric field where dd is greater than the calculated number of digits. dd must be a two-digit number from 01 to 31.
- Udd - use dd digits for the numeric field where dd is greater than or less than the calculated number of digits. dd must be a two-digit number from 01 to 31.
- /x - used to divide numeric data before formatting. /x indicates a division factor as follows:
  - /D: 10
  - /C: 100
  - /K: 1000
  - /KB: 1024
  - /DK: 10\*1000

- /CK: 100\*1000
  - /M: 1000\*1000
  - /MB: 1024\*1024
  - /G: 1000\*1000\*1000
  - /GB: 1024\*1024\*1024
- (p,m,HEX) gives the position and length of a character field to be printed in hexadecimal format. A field must not extend beyond position 32752 or the end of the record.
  - VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record. The following formatting items can be used with VLEN: mask, E'pattern', L'string', F'string', T'string', LZ, NOST, Ndd, Udd and /x.
  - NUM specifies that the record number is to be printed starting at 1 and incrementing by 1 for each record. The following formatting items can be used with NUM: mask, E'pattern', L'string', F'string', T'string', LZ, Ndd and Udd.
  - A description of each type of field is given below.

Format Code	Length	Description
BI	1-8 bytes	Unsigned binary
FI	1-8 bytes	Signed fixed-point
PD	1-16 bytes	Signed packed decimal
ZD	1-31 bytes	Signed zoned decimal
CH	1-1500 bytes	Character
CSF/FS	1-32 bytes	Floating sign
UFF	1-44 bytes	Unsigned free form numeric
SFF	1-44 bytes	Signed free form numeric
DT1	4 bytes	SMF date as Z'yyyymmdd'
DT2	4 bytes	SMF date as Z'yyyymm'
DT3	4 bytes	SMF date as Z'yyyyddd'
DC1	8 bytes	TOD date as Z'yyyymmdd'
DC2	8 bytes	TOD date as Z'yyyymm'
DC3	8 bytes	TOD date as Z'yyyyddd'
DE1	8 bytes	ETOD date as Z'yyyymmdd'
DE2	8 bytes	ETOD date as Z'yyyymm'
DE3	8 bytes	ETOD date as Z'yyyyddd'
TM1	4 bytes	SMF time as Z'hmmss'
TM2	4 bytes	SMF time as Z'hhmm'
TM3	4 bytes	SMF time as Z'hh'
TM4	4 bytes	SMF time as Z'hmmssxx'
TC1	8 bytes	TOD time as Z'hmmss'
TC2	8 bytes	TOD time as Z'hhmm'
TC3	8 bytes	TOD time as Z'hh'

Format Code	Length	Description
TC4	8 bytes	TOD time as Z'hhmmssxx'
TE1	8 bytes	ETOD time as Z'hhmmss'
TE2	8 bytes	ETOD time as Z'hhmm'
TE3	8 bytes	ETOD time as Z'hh'
TE4	8 bytes	ETOD time as Z'hhmmssxx'
HEX	1-1000 bytes	Character printed as hexadecimal
VLEN	n/a	Record length for VLR (1,2,BI)
NUM	n/a	Relative record number

## Optional Operands

- TITLE, PAGE, DATE, DATENS, YDDD, YDDDNS, TIME and TIMENS - the elements to appear in the title line at the top of each page. Only specified elements appear and in the order given. The title string can be 1 to 50 characters. Alternate formats for date and time can be specified.
- HEADER - a one, two or three line heading to be printed for the corresponding ON field (overriding the "standard" one line heading of "(p,m,f)"). A heading string can be 1 to 50 characters. NONE can be used to suppress the heading for the corresponding ON field.
- NOHEADER - suppresses the heading line.
- LINES - the number of lines per page (overriding the default of 58). n can be 10 to 999.
- BLANK and PLUS - causes the column widths to be dynamically adjusted as needed (overriding the standard fixed column widths) and suppresses leading zeros for numeric fields. BLANK causes a blank, rather than a + to be used for the positive sign.
- TOTAL, MAXIMUM, MINIMUM and AVERAGE - the overall statistics to appear for numeric fields after the columns of data for the report. Only the specified statistics appear and in the order given. Each string can be 1 to 50 characters.
- LIMIT - a limit for the number of invalid decimal values (overriding the default of 200). If n invalid decimal values are for an explanation of invalid decimal values.
- VSAMTYPE - the record format for a VSAM input data set (F or V).
- WIDTH - the line length and LRECL for the list data set. n can be 121 to 2048. If the WIDTH specified is less than the calculated line length, ICETOOL terminates the operation. If WIDTH is not specified, the calculated line length (subject to a minimum of 121) is used as the line length and LRECL for the list data set.
- BREAK - the break field to be used to divide the report into sections. Each set of sequential input records with the same break field value is treated as a section in the report. Each section starts on a new page with its own section title and section statistics. The following formatting items can be used with BREAK: mask, E'pattern', L'string', F'string', T'string', LZ and Udd.
- BTITLE - a string to appear in the section title. The break field and string appear in the section title in the order given. The string can be 1 to 50 characters.
- BTOTAL, BMAXIMUM, BMINIMUM and BVERAGE - the section statistics to appear for numeric fields after the columns of data for each section. Only the specified statistics appear for each section and in the order given. Each string can be 1 to 50 characters.
- INDENT - the number of blanks used to indent the report (overriding the default of 0). n can be 0 to 50.

- BETWEEN - the number of blanks used between the columns of data (overriding the default of 3). n can be 0 to 50.
- STATLEFT - places the statistics strings to the left of the first column of data (overriding the default of placing the strings in the first column).
- UZERO - causes -0 to be treated as unsigned, that is, as +0.
- LISTSDB - uses SDB for the list data set (overrides installation value SDBMSG=NO).
- LISTNOSDB - does not use SDB for the list data set (overrides installation value SDBMSG=YES).

## Symbols

- Symbols for fields can be used instead of p,m,f and p,m in the ON and BREAK operands.
- Symbols for character constants can be used instead of 'string' in the TITLE, HEADER, TOTAL, MAXIMUM, MINIMUM, AVERAGE, BTITLE, BTOTAL, BMAXIMUM, BMINIMUM and BAVERAGE operands.

## Notes

- The control character occupies the first byte of each record in the list data set.
- By default, the first column starts in the second byte of each record in the list data set. INDENT(n) or STATLEFT can be used to override this default.
- By default, three blanks appear between columns in the list data set. BETWEEN(n) can be used to override this default.
- RECFM=FBA is used for the list data set.
- Specifying formatting items or PLUS or BLANK, which can compress the columns of output data, can enable you to include more fields in your report, up to a maximum of 20, if your line length is limited by the character width your printer or display supports.

## Example of TOOLMSG Output for DISPLAY

```
* 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.
* Example: print a tailored report showing values from the
* three IN2 data set ON fields, with title, heading lines,
* maximum and minimum, in the DOUT data set.
  DISPLAY FROM(IN2) LIST(DOUT) BLANK -
  TITLE('IN2 Data Set Report') PAGE DATE TIME -
  HEADER('Store') ON(50,15,CH) -
  HEADER('Profit/(Loss)') ON(26,8,ZD,E1) -
  HEADER('Employees') ON(2,3,ZD) -
  MAXIMUM('Largest') MINIMUM('Smallest')
ICE627I 0 DFSORT CALL 0003 FOR COPY FROM IN2      TO E35 EXIT COMPLETED
ICE603I 0 INFORMATION PRINTED IN DOUT      DATA SET
ICE628I 0 RECORD COUNT: 0000000000000008
ICE602I 0 OPERATION RETURN CODE: 00
```

## Example of List Output for DISPLAY

Here's an example of the output that would appear in DOUT:

```
IN2 Data Set Report      - 1 -          02/28/05          13:15:19
```

Store	Profit/(Loss)	Employees
-----	-----	-----
San Jose	72,345,678	123
Morgan Hill	10,273	71
Palo Alto	(52,766,111)	101
Sunnyvale	92,378,566	166
San Francisco	12	27
San Diego	(64,832,715)	102
Los Altos	43,343,732	123
Gilroy	89,348,399	152
Largest	92,378,566	166
Smallest	(64,832,715)	27

## Example using OUTFIL Lookup and Change with DISPLAY

```
COPY FROM(IN) USING(OUTF)
DISPLAY FROM(TEMP) LIST(EMPCT) BLANK -
  TITLE('Employees by Function') -
  YDDD(4D-) -
  HEADER(, 'Function') HEADER('Number of', 'Employees') -
  ON(1,25,CH)          ON(30,4,ZD)
```

OUTFCNTL contains:

```
OUTFIL FNames=TEMP,
  OUTREC=(1:9,2,CHANGE=(25,
    C'MN',C'Manufacturing',
    C'RD',C'Research and Development',
    C'FN',C'Finance',
    C'MR',C'Marketing',
    C'IS',C'Information Systems'),
  30:4,4)
```

## Example of Lookup and Change with DISPLAY Output

Below is an example of the output that would appear in EMPCT.

```
Employees by Function      2005-059

Function                    Number of
-----                    -
Manufacturing              486
Marketing                  21
Research and Development   55
Information Systems        123
Finance                    33
```

Note that the 2-character division codes at position 9 have been replaced by meaningful phrases using DFSORT's OUTFIL lookup and change feature. Complete details on this feature of OUTFIL as well as its many other features can be found in *DFSORT Application Programming Guide*.



## Example of Plain and Fancy Reports

\* Produce a plain report

```
DISPLAY FROM(ACCTS) LIST(PLAIN) -  
  TITLE('Accounts Report for First Quarter') -  
  DATE(MD4/) BLANK -  
  HEADER('Amount') ON(12,6,ZD) -  
  HEADER('Id') ON(NUM) -  
  HEADER('Acct#') ON(31,3,PD) -  
  HEADER('Date') ON(1,4,ZD) -  
  TOTAL('Total for Q1') -  
  AVERAGE('Average for Q1')
```

\* Produce a fancy report by using INDENT,

\* BETWEEN, STATLEFT and formatting items

\* to improve its appearance

```
DISPLAY FROM(ACCTS) LIST(FANCY) -  
  TITLE('Accounts Report for First Quarter') -  
  DATE(MD4/) BLANK -  
  HEADER('Amount') ON(12,6,ZD,C1,N08) -  
  HEADER('Id') ON(NUM,N02) -  
  HEADER('Acct#') ON(31,3,PD,NOST,LZ) -  
  HEADER('Date') ON(1,4,ZD,E'99/99',NOST) -  
  INDENT(2) BETWEEN(5) -  
  STATLEFT -  
  TOTAL('Total for Q1') -  
  AVERAGE('Average for Q1')
```

## Example of Plain Report Output

Accounts Report for First Quarter

04/18/2004

Amount	Id	Acct#	Date
93271	1	15932	106
137622	2	187	128
83147	3	15932	212
183261	4	2158	217
76389	5	187	305
920013	6	15932	319
Total for Q1			
1493703		50328	1287
Average for Q1			
248950		8388	214

## Example of Fancy Report Output

Accounts Report for First Quarter

04/18/2004

Amount	Id	Acct#	Date
932.71	1	15932	01/06
1,376.22	2	00187	01/28
831.47	3	15932	02/12
1,832.61	4	02158	02/17
763.89	5	00187	03/05
9,200.13	6	15932	03/19
Total for Q1			
14,937.03			
Average for Q1			
2,489.50			

---

## MODE Operator Details

### Syntax

```
MODE STOP
      CONTINUE
      SCAN
```

### Function

Specifies one of three modes to control error checking and actions after error detection. A MODE operator affects the processing of the group of operators which follow it, up to the next MODE operator (if any). Dependent operators (those for which a failure of one should stop execution of the rest) can be grouped with MODE STOP. Independent operators (those for which a failure of one should not affect execution of the rest) can be grouped with MODE CONTINUE. Operators to be checked for errors only can be grouped with MODE SCAN.

### Examples

```
MODE CONTINUE
<independent operator group>
```

```
MODE STOP
<dependent operator group>
```

### Required Operand

STOP, CONTINUE or SCAN.

- STOP - If an operation fails, stops processing the remaining operators in the group, but continues to check for errors in ICETOOL statements. STOP mode is the default set at the beginning of an ICETOOL run.
- CONTINUE - If an operation fails, continues processing the remaining operators in the group.
- SCAN - Checks for errors in ICETOOL statements without processing the operators. Set automatically if an error is detected while in STOP mode.

### Note

- The return codes for one group of operators does not affect the other groups of operators.

### Example of TOOLMSG Output for MODE

```
* MODE - specifies the error checking and actions after error
* detection to be performed for a group of operators.
* Example: continue processing operators whether or not an
* error is detected.
      MODE CONTINUE
ICE630I 1 MODE IN EFFECT: CONTINUE
ICE602I 1 OPERATION RETURN CODE: 00
```

---

## OCCUR Operator Details

### Syntax

```
OCCUR FROM(indd) LIST(listdd) ON(p,m,f) ...
OCCURS                                ON(p,m,f,formatting)
                                       ON(p,m,HEX)
                                       ON(VLEN)
                                       ON(VLEN,formatting)
                                       ON(VALCNT)
                                       ON(VALCNT,formatting)

TITLE('string') PAGE DATE            TIME
                                DATE(abcd) TIME(abc)
                                DATENS(abc) TIMENS(ab)
                                YDDD(abc)
                                YDDDNS(ab)

HEADER('string1')                ... LINES(n) BLANK
HEADER('string1','string2')      PLUS
HEADER('string1','string2','string3')
HEADER(NONE)
NOHEADER

ALLDUPS    VSAMTYPE(x) WIDTH(n)
NODUPS
HIGHER(x)
LOWER(y)
EQUAL(v)

INDENT(n) BETWEEN(n) UZERO LISTSDB
                                LISTNOSDB
```

### Function

Prints, in the listdd data set, each unique value for specified numeric (including SMF, TOD and ETOD date/time) and/or character fields and how many times it occurs. From 1 to 10 ON fields can be specified as long as the resulting list data set line length does not exceed the limit specified by the WIDTH operand or 2048 bytes. All ON fields specified are used to determine whether a record contains a unique value. A single list data set record is printed for each unique value with the fields printed in columns in the same order in which they are specified.

Simple or tailored reports can be produced.

- A title line consisting of a string, the page number, the date and/or the time, in any order specified, can appear at the top of each page.
- One, two or three line headings can be printed to identify each specified field.
- Column widths are adjusted automatically according to the operands specified and the length of the fields.
- Alternate formats for date and time can be specified.
- Numeric fields, consisting of a sign and up to 31 digits in various formats, can be edited with separators, decimal points, patterns, and leading, trailing and floating signs.

The ON values printed can be limited to only duplicates, non-duplicates or those that occur less than, equal to or more than n times.

## Examples

```
OCCUR FROM(SOURCE) LIST(VOLSERS) ON(40,6,CH) ON(VALCNT)
```

```
OCCUR FROM(FAILURES) LIST(CHECKIT) LISTNOSDB -
DATE(YMD.) TITLE('Possible System Intruders') PAGE -
HEADER(' Userid ') ON(23,8,CH) -
HEADER(' Logon Failures ') ON(VALCNT) -
HIGHER(4) BLANK
```

## Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- LIST - the ddname of the list data set. You must supply a DD statement for the ddname you specify.
- ON - a field to be used for this operation. From 1 to 10 ON fields can be specified.
  - (p,m,f) gives the position, length and format of a numeric or character field. A field must not extend beyond position 32752 or the end of the record.
  - (p,m,f,formatting) gives the position, length and format of a numeric or character field and specifies how the data for this field is to be formatted for printing. A field must not extend beyond position 32752 or the end of the record.

The following formatting items can be used (see *DFSORT Application Programming Guide* for complete details):

- Mask - an edit mask to be applied to the numeric data. Thirty-nine pre-defined edit masks are available, encompassing many of the numeric notations throughout the world with respect to separators, decimal point, decimal places, signs and so on. See "Appendix A" for complete descriptions and examples of all thirty-nine masks. The attributes of each group of masks is shown below.

Masks	Separators	Decimal Places	Positive Sign	Negative Sign
A0	No	0	blank	-
A1-A5	Yes	0	blank	-
B1-B6	Yes	1	blank	-
C1-C6	Yes	2	blank	-
D1-D6	Yes	3	blank	-
E1-E4	Yes	0	blank	( )
F1-F5	Yes	2	blank	( )
G1-G6	Yes	4	blank	-

- E'pattern' - an edit pattern (1 to 44 characters) to be applied to the numeric data. Each **9** in the pattern (up to 31) is replaced by a corresponding digit (0-9) from the numeric value. Other characters in the pattern appear as specified. E'pattern' can be used for formatting unsigned numeric data such as telephone numbers, dates, and so on.
- L'string' - a leading string to appear at the beginning of the character or numeric data column. Each string can be 1 to 10 characters.
- F'string' - a floating string to appear to the left of numeric data. Each string can be 1 to 10 characters.

- T'string' - a trailing string to appear at the end of the character or numeric data column. Each string can be 1 to 10 characters.
- LZ - used with an edit mask to print leading zeros for the numeric field (overriding the default of suppressing leading zeros).
- Ndd - use dd digits for the numeric field where dd is greater than the calculated number of digits. dd must be a two-digit number from 01 to 31.
- Udd - use dd digits for the numeric field where dd is greater than or less than the calculated number of digits. dd must be a two-digit number from 01 to 31.
- (p,m,HEX) gives the position and length of a character field to be printed in hexadecimal format. A field must not extend beyond position 32752 or the end of the record.
- VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record. The following formatting items can be used with VLEN: mask, E'pattern', L'string', F'string', T'string', LZ, Ndd and Udd.
- VALCNT causes the count of occurrences for each value to be printed. The following formatting items can be used with VALCNT: mask, E'pattern', L'string', F'string', T'string', LZ, Ndd and Udd.
- A description of each type of field is given below.

<b>Format Code</b>	<b>Length</b>	<b>Description</b>
BI	1-8 bytes	Unsigned binary
FI	1-8 bytes	Signed fixed-point
PD	1-16 bytes	Signed packed decimal
ZD	1-31 bytes	Signed zoned decimal
CH	1-1500 bytes	Character
CSF/FS	1-32 bytes	Floating sign
UFF	1-44 bytes	Unsigned free form numeric
SFF	1-44 bytes	Signed free form numeric
DT1	4 bytes	SMF date as Z'yyyymmdd'
DT2	4 bytes	SMF date as Z'yyyymm'
DT3	4 bytes	SMF date as Z'yyyddd'
DC1	8 bytes	TOD date as Z'yyyymmdd'
DC2	8 bytes	TOD date as Z'yyyymm'
DC3	8 bytes	TOD date as Z'yyyddd'
DE1	8 bytes	ETOD date as Z'yyyymmdd'
DE2	8 bytes	ETOD date as Z'yyyymm'
DE3	8 bytes	ETOD date as Z'yyyddd'
TM1	4 bytes	SMF time as Z'hmmss'
TM2	4 bytes	SMF time as Z'hmm'
TM3	4 bytes	SMF time as Z'hh'
TM4	4 bytes	SMF time as Z'hmmssxx'
TC1	8 bytes	TOD time as Z'hmmss'
TC2	8 bytes	TOD time as Z'hmm'

Format Code	Length	Description
TC3	8 bytes	TOD time as Z'hh'
TC4	8 bytes	TOD time as Z'hhmmssxx'
TE1	8 bytes	ETOD time as Z'hhmmss'
TE2	8 bytes	ETOD time as Z'hhmm'
TE3	8 bytes	ETOD time as Z'hh'
TE4	8 bytes	ETOD time as Z'hhmmssxx'
HEX	1-1000 bytes	Character printed as hexadecimal
VLEN	n/a	Record length for VLR (1,2,BI)
VALCNT	n/a	Value count

## Optional Operands

- TITLE, PAGE, DATE, DATENS, YDDD, YDDDNS, TIME and TIMENS - the elements to appear in the title line at the top of each page. Only specified elements appear and in the order given. The title string can be 1 to 50 characters. Alternate formats for date and time can be specified.
- HEADER - a one, two or three line heading to be printed for the corresponding ON field (overriding the "standard" one line heading of "(p,m,f)"). A heading string can be 1 to 50 characters. NONE can be used to suppress the heading for the corresponding ON field.
- NOHEADER - suppresses the heading line.
- LINES - the number of lines per page (overriding the default of 58). n can be 10 to 999.
- BLANK and PLUS - causes the column widths to be dynamically adjusted as needed (overriding the standard fixed column widths) and suppresses leading zeros for numeric fields. BLANK causes a blank, rather than a + to be used for the positive sign.
- ALLDUPS, NODUPS, HIGHER, LOWER, EQUAL - limits the values to be printed to those whose occurrences meet the given criteria. x, y, and v must be specified as n or +n where n can be 1 to 15 decimal digits.
- VSAMTYPE - the record format for a VSAM input data set (F or V).
- WIDTH - the line length and LRECL for the list data set. n can be 121 to 2048. If the WIDTH specified is less than the calculated line length, ICETOOL terminates the operation. If WIDTH is not specified, the calculated line length (subject to a minimum of 121) is used as the line length and LRECL for the list data set.
- INDENT - the number of blanks used to indent the report (overriding the default of 0). n can be 0 to 50.
- BETWEEN - the number of blanks used between the columns of data (overriding the default of 3). n can be 0 to 50.
- UZERO - causes -0 to be treated as unsigned, that is, as +0.
- LISTSDDB - uses SDB for the list data set (overrides installation value SDBMSG=NO).
- LISTNOSDB - does not use SDB for the list data set (overrides installation value SDBMSG=YES).

## Symbols

- Symbols for fields can be used instead of p,m,f and p,m in the ON operand.
- Symbols for character constants can be used instead of 'string' in the TITLE and HEADER operands.
- Symbols for decimal constants can be used instead of n and +n in the HIGHER, LOWER and EQUAL operands.

## Notes

- The control character occupies the first byte of each record in the list data set.
- By default, the first column starts in the second byte of each record in the list data set. INDENT(n) can be used to override this default.
- By default, three blanks appear between columns in the list data set. BETWEEN(n) can be used to override this default.
- RECFM=FBA is used for the list data set.
- The DFSORT DYNALLOC option is used to ensure that work space is available for the sort performed for the OCCUR operation.
- Tape work data sets cannot be used with ICETOOL.
- Specifying PLUS or BLANK, which can compress the columns of output data, can enable you to include more fields in your report, up to a maximum of 10, if your line length is limited by the character width your printer or display supports.



## Example of TOOLMSG Output for OCCUR

- \* OCCUR - prints each unique value for specified numeric and
- \* 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 meets specified criteria (e.g. only duplicate values).
- \* Example: print a report showing values from the IN1 data set
- \* ON field, with the number of times each value occurs, in the
- \* LIST1 data set.

```
OCCUR FROM(IN1) LIST(LIST1) TITLE('Dep't Counts') -
ON(35,3,CH) ON(VALCNT)
ICE627I 0 DFSORT CALL 0004 FOR SORT FROM IN1      TO E35 EXIT COMPLETED
ICE603I 0 INFORMATION PRINTED IN LIST1  DATA SET
ICE628I 0 RECORD COUNT: 000000000000072
ICE638I 0 NUMBER OF RECORDS RESULTING FROM CRITERIA: 000000000000005
ICE602I 0 OPERATION RETURN CODE: 00
```

## Example of List Output for OCCUR

Here's an example of the output that would appear in LIST1:

Dep't Counts

(35,3,CH)	VALUE COUNT
D54	000000000000012
J69	000000000000009
J82	000000000000020
L92	000000000000017
M27	000000000000014

## Example of HIGHER with OCCUR

```
OCCUR FROM(CALLERS) LIST(CALLRPT) -
DATENS(4MD) TITLE('Large Call Volume') -
INDENT(5) BETWEEN(10) -
HEADER('Phone Number') ON(7,10,ZD,E'(999)-999-9999')-
HEADER('Calls') ON(VALCNT,A1,N06)-
HIGHER(1000)
```

## Example of HIGHER with OCCUR Output

20050228	Large Call Volume
Phone Number	Calls
-----	-----
(037)-325-1807	3,521
(216)-721-5530	2,100
(856)-003-3008	12,163

---

# RANGE Operator Details

## Syntax

```
RANGE FROM(indd) ON(p,m,f) HIGHER(x) VSAMTYPE(x)
                ON(VLEN) LOWER(y)
                EQUAL(v)
                NOTEQUAL(w)
```

## Function

Prints a message in TOOLMSG containing the count of values in a specified range for a specific numeric field. The range can be specified as higher than x, lower than y, higher than x and lower than y, equal to v, or not equal to w, where x, y, v, and w are signed or unsigned decimal values.

## Examples

```
RANGE FROM(DATA1) ON(VLEN) HIGHER(52)
RANGE FROM(DATA2) ON(31,18,ZD) EQUAL(-123456789012345678)
RANGE FROM(DATA2) ON(25,3,PD) HIGHER(-20) LOWER(+15)
```

## Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- ON - a field to be used for this operation.
  - (p,m,f) gives the position, length and format of a numeric field. A field must not extend beyond position 32752 or the end of the record.
  - VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record.
  - A description of each type of field is given below.

Format Code	Length	Description
BI	1-8 bytes	Unsigned binary
FI	1-8 bytes	Signed fixed-point
PD	1-16 bytes	Signed packed decimal
ZD	1-31 bytes	Signed zoned decimal
CSF/FS	1-32 bytes	Floating sign
UFF	1-44 bytes	Unsigned free form numeric
SFF	1-44 bytes	Signed free form numeric
VLEN	n/a	Record length for VLR (1,2,BI)

- HIGHER, LOWER, EQUAL, NOTEQUAL - defines the range for the values to be counted. HIGHER and LOWER may be used together or separately. EQUAL and NOTEQUAL must be used separately. x, y, v, and w must be specified as n, +n, or -n where n can be 1 to 31 digits.

## Optional Operand

- VSAMTYPE - the record format for a VSAM input data set (F or V).

## Symbols

- Symbols for fields can be used instead of p,m,f and p,m in the ON operand.
- Symbols for decimal constants can be used instead of n, +n and -n in the HIGHER, LOWER, EQUAL and NOTEQUAL operands.

## Note

- If the range is specified as HIGHER(x) LOWER(y), it must be a valid range. For example, HIGHER(5) LOWER(6) is not a valid range since there is no integer value that satisfies the criteria.

## Example of TOOLMSG Output for RANGE

```
* RANGE - prints a message in TOOLMSG containing the count
* of values in a specified range for a specified numeric
* field in a data set.
* Example: print a count of the values in the IN2 data set
* ON field that are higher than -50, but lower than +100.
  RANGE FROM(IN2) ON(20,2,PD) HIGHER(-50) LOWER(100)
ICE627I 0 DFSORT CALL 0005 FOR COPY FROM IN2      TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 0000000000000008
ICE631I 0 NUMBER OF VALUES IN RANGE FOR (20,2,PD) : 0000000000000004
ICE602I 0 OPERATION RETURN CODE: 00
```

---

## SELECT Operator Details

### Syntax

```
SELECT FROM(indd) TO(outdd)      ON(p,m,f) ...
      DISCARD(savedd) ON(VLEN)

      ALLDUPS  VSAMTYPE(x) UZERO USING(xxxx)
      NODUPS
      HIGHER(x)
      LOWER(y)
      EQUAL(v)
      FIRST
      LAST
      FIRSTDUP
      LASTDUP
```

### Function

Selects records from the indd data set for inclusion in the outdd data set based on meeting criteria for the number of times specified numeric and/or character field values occur. From 1 to 10 ON fields can be specified. All ON fields are used to determine the value count (that is, the number of times the ON values occur) to be matched against the criteria.

The records selected can be limited to those with duplicate values, non-duplicate values, values that occur less than, equal to or more than n times, or the first or last record with each unique or duplicate value.

DISCARD(savedd) can be used to save the records which do not meet the criteria (that is, the discarded records), in the savedd data set. DISCARD(savedd) may be used with or without TO(outdd).

The DFSORT control statements in xxxxCNTL are used if USING(yyyy) is specified.

### Examples

```
SELECT FROM(INPUT) TO(DUPS) ON(11,8,CH) ON(30,44,CH) ALLDUPS
```

```
SELECT FROM(IN) TO(OUT1) ON(29,5,FS) HIGHER(3)
```

```
SELECT FROM(DATA) TO(NEWEST) ON(135,20,ZD) LASTDUP USING(CTL1)
```

```
SELECT FROM(BOOKS) TO(PUBLISHR) ON(29,10,UFF) FIRST -
      DISCARD(SAVEREST)
```

### Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- TO - the ddname of the output data set for the selected records. You must supply a DD statement for the ddname you specify.
- DISCARD - the ddname of the output data set for the records which are not selected. You must supply a DD statement for the ddname you specify. TO(outdd) and DISCARD(savedd) may be used together or separately.
- ON - a field to be used for this operation. From 1 to 10 ON fields can be specified.

- (p,m,f) gives the position, length and format of a numeric or character field. A field must not extend beyond position 32752 or the end of the record.
- VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record.
- A description of each type of field is given below.

Format Code	Length	Description
BI	1-1500 bytes	Unsigned binary
FI	1-256 bytes	Signed fixed-point
PD	1-16 bytes	Signed packed decimal
ZD	1-31 bytes	Signed zoned decimal
CH	1-1500 bytes	Character
CSF/FS	1-32 bytes	Floating sign
UFF	1-44 bytes	Unsigned free form numeric
SFF	1-44 bytes	Signed free form numeric
VLEN	n/a	Record length for VLR (1,2,BI)

- ALLDUPS, NODUPS, HIGHER, LOWER, EQUAL, FIRST, LAST, FIRSTDUP, LASTDUP - defines the criteria against which the value counts are to be matched. x, y, and v must be specified as n or +n where n can 0 to 99.

## Optional Operands

- VSAMTYPE - the record format for a VSAM input data set (F or V).
- UZERO - causes -0 to be treated as unsigned, that is, as +0.
- USING - the first 4 characters of the ddname (xxxxCNTL) for the DFSORT control statement data set. You must supply a DD statement for xxxxCNTL if you specify USING(xxxx).

If used, the xxxxCNTL data set should only contain one or more of these statements:

- INCLUDE or OMIT - deleted records are not processed by SELECT.
- OPTION - for optional parameters like MAINSIZE.
- OUTFIL - the selected records are processed by OUTFIL. With TO(outdd), use a corresponding OUTFIL statement with FNAMES=outdd. With DISCARD(savedd), use a corresponding OUTFIL statement with FNAMES=savedd.

## Symbols

- Symbols for fields can be used instead of p,m,f and p,m in the ON operand.
- Symbols for decimal constants can be used instead of n and +n in the HIGHER, LOWER and EQUAL operands.

## Notes

- The DFSORT DYNALLOC option is used to ensure that work space is available for the sort performed for the SELECT operation.
- Tape work data sets cannot be used with ICETOOL.

## Example of TOOLMSG Output for SELECT

- \* 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 (e.g. only duplicate values).
- \* Records that are not selected can be saved in a separate
- \* output data set.
- \* Example: selects records from the IN1 data set, for the
- \* SEL1 data set, whose ON field occurs only once (i.e, only
- \* records with no duplicate ON field values).

SELECT FROM(IN1) TO(SEL1) ON(30,2,PD) NODUPS

ICE627I 0 DFSORT CALL 0006 FOR SORT FROM IN1 TO SEL1 COMPLETED

ICE628I 0 RECORD COUNT: 000000000000072

ICE638I 0 NUMBER OF RECORDS RESULTING FROM CRITERIA: 000000000000013

ICE602I 0 OPERATION RETURN CODE: 00

---

## SORT Operator Details

### Syntax

```
SORT FROM(indd) USING(xxxx) TO(outdd,...) VSAMTYPE(x)
```

```
    LOCALE(name)    SERIAL  
    LOCALE(CURRENT)  
    LOCALE(NONE)
```

### Function

Sorts the indd data set to the outdd data sets (up to 10) using the the DFSORT control statements in xxxxCNTL. You must supply a DFSORT SORT statement in xxxxCNTL to indicate the control fields for the sort. Additional DFSORT control statements and options can be used to sort a subset of the input records (INCLUDE or OMIT statement; SKIPREC and STOPAFT options; OUTFIL INCLUDE, OMIT, STARTREC, ENDREC and SPLIT operands), reformat records for output (INREC, OUTREC and OUTFIL statements), and so on.

### Examples

```
SORT FROM(A) TO(B,C,D) USING(DEPT)
```

```
SORT FROM(VSAMIN) TO(VSAMOUT1) USING(VSAM) -  
    VSAMTYPE(F) LOCALE(DA_DK)
```

```
SORT FROM(MASTER) USING(MULT)
```

DEPTCNTL might contain the following:

```
//DEPTCNTL DD *  
    SORT FIELDS=(15,3,A,27,4,D),FORMAT=PD  
    OUTREC FIELDS=(1,10,2Z,15,3,3Z,27,4)  
/*
```

VSAMCNTL might contain the following:

```
//VSAMCNTL DD *  
    SORT FIELDS=(27,12,CH,A)  
/*
```

MULTCNTL might contain the following:

```
//MULTCNTL DD *  
    SORT FIELDS=(18,3,ZD,A,43,2,BI,D)  
    OUTFIL FNames=DEPT1,INCLUDE=(5,3,CH,EQ,C'D01')  
    OUTFIL FNames=DEPT2,INCLUDE=(5,3,CH,EQ,C'D02')  
    OUTFIL FNames=DEPT3,INCLUDE=(5,3,CH,EQ,C'D03')  
    OUTFIL FNames=REST,SAVE  
/*
```

Note that DFSORT's OUTFIL INCLUDE and SAVE features are used here to create four different subset data sets using a single pass over the input data set. SAVE allows you to easily include the records that are not included in any other OUTFIL data set. Complete details on this feature of OUTFIL as well as its many other features can be found in *DFSORT Application Programming Guide*.



## Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- USING - the first 4 characters of the ddname (xxxxCNTL) for the DFSORT control statement data set. You must supply a DD statement for xxxxCNTL.

## Optional Operands

- TO - the ddnames of 1 to 10 output data sets. You must supply DD statements for the ddnames you specify.
- VSAMTYPE - the record format for a VSAM input data set (F or V).
- LOCALE - overrides the installation default for locale processing.
- SERIAL - causes OUTFIL processing not to be used for multiple outdd data sets. SERIAL is not recommended because it imposes data set restrictions and can degrade performance.

## Notes

- If you use OUTFIL statements in xxxxCNTL to specify your output data sets, you do not need to specify TO.
- The DFSORT DYNALLOC option is used to ensure that work space is available for the sort.
- Tape work data sets cannot be used with ICETOOL.

## Example of TOOLMSG Output for SORT

```
* SORT - sorts a data set to one or more output data sets.
* Multiple output is handled using a single pass over
* the input.
* Example: sort the IN1 data set to the OUT1 data set using
* the DFSORT control statements in the CTL1CNTL data set.
  SORT FROM(IN1) TO(OUT1) USING(CTL1)
ICE606I 0 DFSORT CALL 0007 FOR SORT FROM IN1      TO OUT1
        USING CTL1CNTL COMPLETED
ICE602I 0 OPERATION RETURN CODE:  00
```

---

# SPLICE Operator Details

## Syntax

```
SPLICE FROM(indd) TO(outdd) ON(p,m,f) ... WITH(p,m) ...  
  
    WITHEACH KEEPNOUDUPS KEEPBASE VSAMTYPE(x) UZERO USING(yyyy)  
    WITHALL  
  
    VLENMAX  
    VLENOVLY
```

## Function

Splices together specified fields from records with matching numeric or character field values (that is, duplicate values), but different information. This makes it possible to join fields from different types of input records to create an output record with information from two or more records. From 1 to 10 ON fields can be specified. All ON fields are used to determine if records match.

Typically, you will want to reformat the records from two or more data sets to temporary data sets, and concatenate those temporary data sets together as input to the SPLICE operator.

By default, the first duplicate is spliced with all of the WITH fields from the last duplicate. From 1 to 50 WITH fields can be specified. WITHALL can be used to splice the first duplicate with all of the WITH fields from the second and subsequent duplicates. WITHEACH can be used to splice the first duplicate with one WITH field from the second and subsequent duplicates in turn.

By default, non-matching records are not kept. KEEPNOUDUPS can be used to keep non-matching records.

By default, the base record is not kept. KEEPBASE can be used to keep the base record.

By default, for variable-length records, the length of the base record is used as the length of the spliced record. VLENMAX can be used to set the length of the spliced record to the maximum length of the base record or overlay record. VLENOVLY can be used to set the length of the spliced record to the length of the overlay record.

The DFSORT control statements in yyyyCNTL are used if USING(yyyy) is specified.

## Examples

```
SPLICE FROM(CON1) TO(OUT1) ON(11,8,CH) ON(30,44,CH) WITH(1,100)
```

```
SPLICE FROM(CON1) TO(OUT2) ON(11,8,CH) -  
    WITHALL WITH(51,50) WITH(101,75) -  
    KEEPNOUDUPS USING(CTL1)
```

```
SPLICE FROM(CON2) TO(OUT3) ON(15,5,ZD) -  
    WITHEACH WITH(21,20) WITH(41,20) WITH(61,20)
```

## Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- TO - the ddname of the output data set for the spliced records. You must supply a DD statement for the ddname you specify.
- ON - a field to be used for this operation. From 1 to 10 ON fields can be specified.
  - (p,m,f) gives the position, length and format of a numeric or character field. A field must not extend beyond position 32752 or the end of the record.
  - A description of each type of field is given below.

Format Code	Length	Description
BI	1-1500 bytes	Unsigned binary
FI	1-256 bytes	Signed fixed-point
PD	1-16 bytes	Signed packed decimal
ZD	1-31 bytes	Signed zoned decimal
CH	1-1500 bytes	Character
CSF/FS	1-32 bytes	Floating sign
UFF	1-44 bytes	Unsigned free form numeric
SFF	1-44 bytes	Signed free form numeric

- WITH(p,m) - a field to be spliced from the second or subsequent duplicate record to the first duplicate record. From 1 to 50 WITH fields can be specified. (p,m) gives the position and length of the field.

## Optional Operands

- WITHEACH - the first duplicate is spliced with one specified WITH field from each subsequent duplicate (overriding the default of splicing the first duplicate with all of the specified WITH fields from the last duplicate).
- WITHALL - the first duplicate is spliced with all of the specified WITH fields from the second duplicate, and then from each subsequent duplicate in turn (overriding the default of splicing the first duplicate with all of the specified WITH fields from the last duplicate).
- KEEPNOUDUPS - non-duplicates are kept (unchanged) along with the spliced records (overriding the default of deleting non-duplicates).
- KEEPBASE - the base records are kept (unchanged) along with the spliced records (overriding the default of deleting the base records).
- VSAMTYPE - the record format for a VSAM input data set (F or V).
- UZERO - causes -0 to be treated as unsigned, that is, as +0.
- USING - the first 4 characters of the ddname (xxxxCNTL) for the DFSORT control statement data set. You must supply a DD statement for xxxxCNTL if you specify USING(xxxx).

If used, the xxxxCNTL data set should only contain one or more of these statements:

- INCLUDE or OMIT - deleted records are not processed by SPLICE.
- OPTION - for optional parameters like MAINSIZE.

- OUTFIL - the selected records (spliced records, and unspliced records if KEEPNOUDUPS is specified) are processed by OUTFIL. With TO(outdd), use a corresponding OUTFIL statement with FNames=outdd.
- VLENMAX - for variable-length records, the length of the spliced record is set to the maximum length of the base record or overlay record (overriding the default of setting the length of the spliced record to the length of the base record).
- VLENOVLY - for variable-length records, the length of the spliced record is set to the length of the overlay record (overriding the default of setting the length of the spliced record to the length of the base record).

## Symbols

- Symbols for fields can be used instead of p,m,f and p,m in the ON operand and instead of p,m in the WITH operand.

## Notes

- The DFSORT DYNALLOC option is used to ensure that work space is available for the sort performed for the SPLICE operation.
- Tape work data sets cannot be used with ICETOOL.

## Example of TOOLMSG Output for SPLICE

- \* SPLICE - splices together specified fields from records
- \* that have the same specified numeric or character field
- \* values (i.e., 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 various "join" and "match" operations can
- \* be performed.
- \* Example: for ON fields that occur more than once (i.e.,
- \* duplicate ON field values) in the CONCT data set, splices
- \* the WITH field from the last duplicate record into the
- \* first duplicate record.

```
SPLICE FROM(CONCT) TO(COMBINE) ON(11,5,CH) WITH(41,20)
ICE627I 0 DFSORT CALL 0008 FOR SORT FROM CONCT TO COMBINE COMPLETED
ICE628I 0 RECORD COUNT: 000000000000019
ICE638I 0 NUMBER OF RECORDS RESULTING FROM CRITERIA: 00000000000007
ICE602I 0 OPERATION RETURN CODE: 00
```

---

# STATS Operator Details

## Syntax

```
STATS FROM(indd) ON(p,m,f) VSAMTYPE(x) LMSG  
ON(VLEN)
```

## Function

Prints messages in TOOLMSG containing the minimum, maximum, average and total for up to 10 specified numeric fields. The average is calculated by dividing the total by the record count and rounding down to the nearest integer.

## Examples

```
STATS FROM(DATA1) ON(VLEN) ON(15,4,ZD)
```

```
STATS FROM(VSAMIN) ON(5,8,BI) ON(20,2,PD) ON(12,6,FS) ON(47,3,FI) -  
VSAMTYPE(F)
```

## Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- ON - a field to be used for this operation.
  - (p,m,f) gives the position, length and format of a numeric field. A field must not extend beyond position 32752 or the end of the record.
  - VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record.
  - A description of each type of field is given below.

Format Code	Length	Description
BI	1-8 bytes	Unsigned binary
FI	1-8 bytes	Signed fixed-point
PD	1-16 bytes	Signed packed decimal
ZD	1-31 bytes	Signed zoned decimal
CSF/FS	1-32 bytes	Floating sign
UFF	1-44 bytes	Unsigned free form numeric
SFF	1-44 bytes	Signed free form numeric
VLEN	n/a	Record length for VLR (1,2,BI)

## Optional Operand

- VSAMTYPE - the record format for a VSAM input data set (F or V).
- LMSG - the minimum, maximum, average and total for all numeric fields are printed using messages that display 31 digits (overriding the default of printing messages that display 15 digits when possible).

## Symbols

- Symbols for fields can be used instead of p,m,f and p,m in the ON operand.

## Note

- If the total for a field overflows, ICETOOL continues processing, but prints asterisks for the average and total for that field.

## Example of TOOLMSG Output for STATS

```
* STATS - prints messages in TOOLMSG containing the minimum,
* maximum, average, and total for specified numeric fields in
* a data set.
* Example: print the minimum, maximum, average and total
* values for the three VLRIN data set ON fields.
* For variable-length records, ON(VLEN) gives statistics
* about the length of the records.
  STATS FROM(VLRIN) ON(VLEN) ON(12,2,ZD) ON(18,5,FS)
ICE627I 0 DFSORT CALL 0009 FOR COPY FROM VLRIN   TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000017
ICE607I 0 STATISTICS FOR (VLEN)           :
ICE608I 0   MINIMUM: +000000000000058, MAXIMUM: +000000000000079
ICE609I 0   AVERAGE: +000000000000068, TOTAL  : +0000000000001171
ICE607I 0 STATISTICS FOR (12,2,ZD)       :
ICE608I 0   MINIMUM: -000000000000064, MAXIMUM: +000000000000082
ICE609I 0   AVERAGE: +000000000000010, TOTAL  : +000000000000177
ICE607I 0 STATISTICS FOR (18,5,FS)       :
ICE608I 0   MINIMUM: -000000000003892, MAXIMUM: +000000000018723
ICE609I 0   AVERAGE: +000000000001127, TOTAL  : +000000000019168
ICE602I 0 OPERATION RETURN CODE: 00
```

---

## UNIQUE Operator Details

### Syntax

```
UNIQUE FROM(indd) ON(p,m,f) VSAMTYPE(x) UZERO  
ON(VLEN)
```

### Function

Prints a message in TOOLMSG containing the count of unique values for a specified numeric or character field.

### Examples

```
UNIQUE FROM(INPUT) ON(20,40,CH)
```

```
UNIQUE FROM(DATA) ON(5,3,ZD)
```

### Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- ON - a field to be used for this operation.
  - (p,m,f) gives the position, length and format of a numeric field. A field must not extend beyond position 32752 or the end of the record.
  - VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record.
  - A description of each type of field is given below.

Format Code	Length	Description
BI	1-256 bytes	Unsigned binary
FI	1-256 bytes	Signed fixed-point
PD	1-32 bytes	Signed packed decimal
ZD	1-32 bytes	Signed zoned decimal
CH	1-1500 bytes	Character
CSF/FS	1-32 bytes	Floating sign
UFF	1-44 bytes	Unsigned free form numeric
SFF	1-44 bytes	Signed free form numeric
VLEN	n/a	Record length for VLR (1,2,BI)

### Optional Operands

- VSAMTYPE - the record format for a VSAM input data set (F or V).
- UZERO - causes -0 to be treated as unsigned, that is, as +0.

### Symbols

- Symbols for fields can be used instead of p,m,f and p,m in the ON operand.



## Notes

- The DFSORT DYNALLOC option is used to ensure that work space is available for the sort performed for the UNIQUE operation.
- Tape work data sets cannot be used with ICETOOL.

## Example of TOOLMSG Output for UNIQUE

```
* UNIQUE - prints a message in TOOLMSG containing the count of
* unique values for a specified numeric or character field.
* Example: print the count of unique values in the
* OUT1 data set ON field.
  UNIQUE FROM(OUT1) ON(30,2,PD)
ICE627I 0 DFSORT CALL 0010 FOR SORT FROM OUT1      TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 0000000000000040
ICE610I 0 NUMBER OF UNIQUE VALUES FOR (30,2,PD)   : 000000000000011
ICE602I 0 OPERATION RETURN CODE: 00
```

---

## VERIFY Operator Details

### Syntax

```
VERIFY FROM(indd) ON(p,m,f) NOSIGN LIMIT(n) VSAMTYPE(x)
```

### Function

Examines up to 10 specified decimal fields in a data set and prints a message in TOOLMSG identifying each invalid value found for each field. A decimal value is considered invalid under one of the following circumstances:

- it contains A-F as a digit (example: a PD field of 00AF)
- it contains 0-9 as a sign and the NOSIGN operand is not specified (example: a ZD field of F235).

### Examples

```
VERIFY FROM(NEW) ON(22,16,PD) ON(7,9,PD)
```

```
VERIFY FROM(DATA) ON(28,5,PD) ON(28000,18,ZD) ON(4,7,PD) -  
NOSIGN LIMIT(10)
```

### Required Operands

- FROM - the ddname of the input data set. You must supply a DD statement for the ddname you specify.
- ON - a field to be used for this operation.
  - (p,m,f) gives the position, length and format of a numeric field. A field must not extend beyond position 32752 or the end of the record.
  - A description of each type of field is given below.

Format Code	Length	Description
PD	1-16 bytes	Signed packed decimal
ZD	1-18 bytes	Signed zoned decimal

### Optional Operands

- NOSIGN - the sign of the decimal values is not to be checked for validity.
- LIMIT - a limit for the number of invalid decimal values (overriding the default of 200). If n invalid decimal values are found, ICETOOL terminates the operation.
- VSAMTYPE - the record format for a VSAM input data set (F or V).

### Symbols

- Symbols for fields can be used instead of p,m,f and p,m in the ON operand.

## Notes

- Values with invalid digits are also identified for the DISPLAY, OCCUR, RANGE, and STATS operators.
- For each invalid digit found, ICETOOL identifies the relative record number in which the field appears and the value of the field (in hexadecimal).
- The DISPLAY operator can be used to print a report identifying the relative record number, hexadecimal value and associated fields for each invalid (and valid) decimal value. See *DFSORT Application Programming Guide*, under "DISPLAY Operator", for an example.

## Example of TOOLMSG Output for VERIFY

```
* VERIFY - examines specified decimal fields in a data set and
* prints a message in TOOLMSG identifying each invalid value
* found for each field.
* Example: identify all values in the two IN2 data set
* decimal ON fields that have invalid digits (A-F)
* and/or invalid signs (0-9).
  VERIFY FROM(IN2) ON(10,2,ZD) ON(41,6,PD)
ICE618A 0 INVALID (10,2,ZD)      VALUE - RECORD: 000000000000003,
        HEX VALUE  FAF2
ICE618A 0 INVALID (10,2,ZD)      VALUE - RECORD: 000000000000006,
        HEX VALUE  F134
ICE618A 0 INVALID (41,6,PD)      VALUE - RECORD: 000000000000007,
        HEX VALUE  000000105739
ICE627I 0 DFSORT CALL 0011 FOR COPY FROM IN2      TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000008
ICE602I 0 OPERATION RETURN CODE: 12
```

---

## Symbol Processing

A DFSORT symbol is a name (preferably something meaningful) that represents a field or a constant. Sets of symbols, also called mappings, can be used to describe a group of related fields and constants such as the information in a particular type of record. Such mappings allow you to refer to fields and constants by their symbols, freeing you from having to know the position, length and format of a field or the value of a constant you want to use.

DFSORT's symbol processing feature gives you a powerful, simple and flexible way to create symbol mappings for your own frequently used data. In addition, you can obtain IBM-created symbol mappings and sample jobs for data associated with RACF, DFSMSrmm and DCOLLECT. For details, visit the DFSORT home page at URL:

<http://www.ibm.com/storage/dfsor>

DFSORT symbols can be up to 50 characters, are case-sensitive and can include underscores and hyphens. Thus, you can create meaningful, descriptive names for your symbols, such as Price\_of\_Item (or Price-of-Item), making them easy to remember, use and understand.

You can define and use a symbol for any field or constant in the following ICETOOL operators: COUNT, DISPLAY, OCCUR, RANGE, SELECT, SPLICE, STATS, UNIQUE and VERIFY. You can also use symbols in the DFSORT control statements you specify for an ICETOOL run. Two DD statements are used for symbol processing as follows:

- SYMNames: activates symbols processing and defines the symbol data sets to be used.
- SYMnOut: can be used to define a data set in which the original symbols and the symbol table built by ICETOOL will be displayed.

Here's a sample ICETOOL job that shows how symbols can be used for ICETOOL operators and DFSORT control statements. See *DFSORT Application Programming Guide* for a complete explanation of DFSORT Symbols.

```
//TOOLSymb JOB ...
//DOIT EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=*
//DFSMSG DD SYSOUT=*
//SYMNames DD DSN=MY.CUSTOMER.SYMBOLS,DISP=SHR
//SYMnOut DD SYSOUT=*
//IN DD DSN=MY.CUSTOMER.INPUT,DISP=SHR
//OUT DD DSN=&O,UNIT=SYSDA,SPACE=(CYL,(5,5),RLSE),
// DISP=(,PASS)
//LIST1 DD SYSOUT=*
//TOOLIN DD *
  RANGE FROM(IN) ON(Customer_Balance) LOWER(Stop_Check)
  SORT FROM(IN) TO(OUT) USING(CTL1)
  DISPLAY FROM(OUT) LIST(LIST1) BLANK WIDTH(133) -
    TITLE(Title) DATE(4MD/) PAGE -
    HEADER(Head1) ON(Customer_Name) -
    HEADER(Head2) ON(Customer_Balance,C1) -
    HEADER(Head3) ON(Customer_Flags,HEX)
/*
//CTL1CNTL DD *
  SORT FIELDS=(Customer_Balance,D,Customer_Name,A)
  INCLUDE COND=((Dept_Code,EQ,Research,OR,
    Dept_Code,EQ,Marketing),
    AND,Customer_Balance,GT,Gift)
```

---

## Calling ICETOOL from a Program

ICETOOL can be called from an assembler program using LINK, ATTACH, or XCTL, and standard linkage conventions.

When all operators have been processed, ICETOOL returns to the calling program with register 15 (R15) set to the highest operation return code encountered.

Two different interfaces are available:

- TOOLIN Interface:

You supply ICETOOL statements in the TOOLIN data set. ICETOOL prints messages in the TOOLMSG data set, but does not return information directly to your program.

- Parameter List Interface

You supply ICETOOL statements in a parameter list. ICETOOL prints messages in the TOOLMSG data set and also returns information in the parameter list for use by your program.

For complete details of ICETOOL's calling program interface, see *DFSORT Application Programming Guide*.



# Appendix A. Edit Masks for DISPLAY and OCCUR Formatting

The table below describes the available masks and shows how the values 12345678 and -1234567 would be printed for each mask. In the pattern:

- **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

Mask	Pattern	12345678	-1234567
A0	wdddddddddddddddddddddddddddd	12345678	-1234567
A1	wd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd	12,345,678	-1,234,567
A2	wd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd	12.345.678	-1.234.567
A3	wd ddd ddd ddd ddd ddd ddd ddd ddd ddd	12 345 678	-1 234 567
A4	wd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd	12'345'678	-1'234'567
A5	d ddd ddd ddd ddd ddd ddd ddd ddd dddx	12 345 678	1 234 567-
B1	wddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,d	1,234,567.8	-123,456.7
B2	wddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,d	1.234.567,8	-123.456,7
B3	wddd ddd ddd ddd ddd ddd ddd ddd ddd,d	1 234 567,8	-123 456,7
B4	wddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,d	1'234'567.8	-123'456.7
B5	wddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,d	1'234'567,8	-123'456,7
B6	ddd ddd ddd ddd ddd ddd ddd ddd ddd,dx	1 234 567,8	123 456,7-
C1	wdd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,dd	123,456.78	-12,345.67
C2	wdd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,dd	123.456,78	-12.345,67
C3	wdd ddd ddd ddd ddd ddd ddd ddd ddd,dd	123 456,78	-12 345,67
C4	wdd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,dd	123'456.78	-12'345.67
C5	wdd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,dd	123'456,78	-12'345,67
C6	dd ddd ddd ddd ddd d ddd ddd ddd ddd,ddx	123 456,78	12 345,67-
D1	wd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd	12,345.678	-1,234.567
D2	wd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,ddd	12.345,678	-1.234,567
D3	wd ddd ddd ddd ddd ddd ddd ddd ddd,ddd	12 345,678	-1 234,567
D4	wd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,ddd	12'345.678	-1'234.567
D5	wd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,ddd	12'345,678	-1'234,567
D6	d ddd ddd ddd ddd ddd ddd ddd ddd,dddx	12 345,678	1 234,567-

Mask	Pattern	12345678	-1234567
E1	yd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,dddz	12,345,678	(1,234,567)
E2	yd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.dddz	12.345.678	(1.234.567)
E3	yd ddd ddd ddd ddd ddd ddd ddd ddd dddz	12 345 678	(1 234 567)
E4	yd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'dddz	12'345'678	(1'234'567)
F1	ydd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddz	123,456.78	(12,345.67)
F2	ydd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,ddz	123.456,78	(12.345,67)
F3	ydd ddd ddd ddd ddd ddd ddd ddd ddd,ddz	123 456,78	(12 345,67)
F4	ydd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,ddz	123'456.78	(12'345.67)
F5	ydd'ddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,ddz	123'456,78	(12'345,67)
G1	wddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd,ddd	1,234.6678	-123.4567
G2	wddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd.ddd,ddd	1.234,5678	-123,4567
G3	wddd ddd ddd ddd ddd ddd ddd ddd,ddd	1 234,5678	-123,4567
G4	wddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,ddd	1'234.5678	-123.4567
G5	wddd'ddd'ddd'ddd'ddd'ddd'ddd'ddd,ddd	1'234,5678	-123,4567
G6	ddd ddd ddd ddd ddd ddd ddd ddd,dddxd	1 234,5678	123,4567-