DFSORT

# Reference Summary

*Release 14*

SX33-8001-14

**Note!**
Before using this information and the product it supports, be sure to read the general information under Chapter 6, “Notices” on page 101.

**Fifteenth Edition (March 2002)**

This edition replaces and makes obsolete the previous edition, SX33-8001-13. The technical changes for this edition are indicated by a vertical bar to the left of a change.

This edition applies to Release 14 of DFSORT, 5740-SM1, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. The information in this summary is compiled from *DFSORT Application Programming Guide* , SC33-4035-21. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address below.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

# Contents



# Preface

## About this Book

This book summarizes the DFSORT program control statements, ICETOOL operators, and job control language (JCL) statements detailed in *DFSORT Application Programming Guide R14*, which should be referred to for detailed information on DFSORT program control statements, ICETOOL operators, and the JCL to be used with DFSORT.

## Required Product Knowledge

To use this book effectively, you should be familiar with the concepts and material presented in *DFSORT Application Programming Guide R14*. *DFSORT Application Programming Guide R14* provides general-use programming interfaces which allow you to write programs using DFSORT. This book assumes you know how to develop DFSORT applications.

You should be familiar with:
- DFSORT program control statements
- ICETOOL operators
- Job control language statements.

## Required Publications

You should be familiar with the information in this publication:

| Publication Title | Order Number |
|---|---|
| *DFSORT Application Programming Guide R14* | SC33–4035 |

## DFSORT Publications

*DFSORT Reference Summary* is a part of a more extensive DFSORT library. These books can help you work with DFSORT more effectively.

| Task | Publication | Order Number |
|---|---|---|
| Planning For and Customizing DFSORT | *DFSORT Installation and Customization R14* | SC33-4034 |
| Learning to Use DFSORT Panels | *DFSORT Panels Guide* | GC26-7037 |
| Learning to Use DFSORT | *DFSORT Getting Started R14* | SC26-4109 |
| Application Programming | *DFSORT Application Programming Guide R14* | SC33–4035 |

| Task | Publication | Order Number |
|---|---|---|
| Diagnosing Failures and Interpreting Messages | *DFSORT Messages, Codes and Diagnosis Guide R14* | SC26-7050 |
| Tuning DFSORT | *DFSORT Tuning Guide R14* | SC26-3111 |

You can order a complete set of DFSORT publications with the order number SBOF-1243, except for *DFSORT Licensed Program Specifications R14*, GC33-4032, which must be ordered separately.

## DFSORT Library Softcopy Information

The DFSORT library is available on CD-ROM.

| Order Number | Title |
|---|---|
| SK3T-4269 | *z/OS Collection* |

## How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other DFSORT documentation:

- Send your comments by e-mail to:
- IBMLink from US: starpubs@us.ibm.com
- IBMLink from Canada: STARPUBS at TORIBM
- IBM Mail Exchange: USIB3VVD at IBMMAIL
- Internet: starpubs@us.ibm.com

Be sure to include the name of the book, the part number of the book, version and product name, and if applicable, the specific location of the text you are commenting on (for example, a page number or table number).

## DFSORT Web Site

For articles, online books, news, tips, techniques, examples, and more, visit the DFSORT/MVS home page at:

`http://www.ibm.com/storage/dfsort/`

## DFSORT FTP Site

You can obtain DFSORT articles and examples via anonymous FTP to:

`ftp.software.ibm.com/storage/dfsort/mvs/`

# Chapter 1. General Coding Information

## Notational Conventions

The syntax diagrams in this book are designed to make coding DFSORT™ program control statements simple and unambiguous. The lines and arrows represent a path or flowchart that connects operators, parameters, and delimiters in the order and syntax in which they must appear in your completed statement. Construct a statement by tracing a path through the appropriate diagram that includes all the parameters you need, and code them in the order that the diagram requires you to follow. Any path through the diagram gives you a correctly coded statement, if you observe these conventions:

- Read the syntax diagrams from left to right and from top to bottom.
- Begin coding your statement at the spot marked with the double arrowhead.



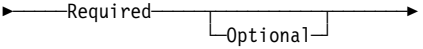
- A single arrowhead at the end of a line indicates that the diagram continues on the next line or at an indicated spot.



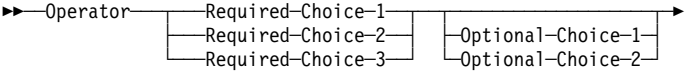
  A continuation line begins with a single arrowhead.



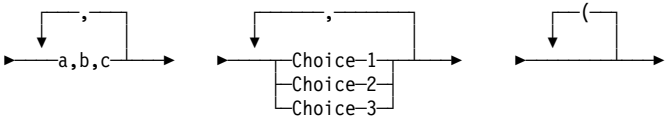
- Strings in upper-case letters and punctuation (parentheses, apostrophes, and so on) must be coded exactly as shown.
  - Semicolons are interchangeable with commas in program control statements and the EXEC PARM string. For clarity, only commas are shown in this book.
- Strings in all lowercase letters represent information that you supply.
- Required parameters appear on the same horizontal line (the main path) as the operator, while optional parameters appear in a branch below the main path.




- Where you can make one choice between two or more parameters, the alternatives are stacked vertically.
  If one choice within the stack lies on the main path (as in the example




  above, left), you *must* specify one of the alternatives. If the stack is placed below the main path (as in the example above, right), then selections are optional, and you can choose either one or none of them.
- The repeat symbol shows where you can return to an earlier position in the syntax diagram to specify a parameter more than once (see the example below, left), to specify more than one choice at a time from the same stack (see the example below, middle), or to nest parentheses (see the example below, right).




  Do not interpret a repeat symbol to mean that you can specify incompatible parameters. For instance, do not specify both **ABEND** and **NOABEND** in the same **DEBUG** statement, or attempt to nest parentheses incorrectly.

  Use any punctuation or delimiters that appear within the repeat symbol to separate repeated items.
- A double arrowhead at the end of a line indicates the end of the syntax diagram.



## DFSORT Control Statement Coding Rules

See *Application Programming Guide* for complete details on coding DFSORT program control statements.

## Inserting Comment Statements

- A statement with an asterisk (*) in column 1 is treated as a comment statement; it is printed along with other DFSORT program control statements but is not otherwise processed.
- A statement with blanks in columns 1 through 71 is treated as a comment statement.
- Comment statements are allowed in the DFSPARM, SYSIN, and SORTCNTL data sets but not in the parameter lists.

## Coding Restrictions

The following rules apply to control statement preparation:

- Labels, operation definers, and operands must be in uppercase EBCDIC.
- Column 1 of each control statement can be used only for a label or for a comment statement that begins with an asterisk in column 1.
- Labels must begin in column 1 and conform to operating system requirements for statement labels.
- The entire operation definer must be contained on the first line of a control statement.
- The first operand must begin on the first line of a control statement. The last operand in a statement must be followed by at least one blank.
- Blanks are not allowed in operands. Anything following a blank is considered part of the remark field.
- In general, values can contain no more than eight alphanumeric characters. Values that specify record counts, such as those for SKIPREC, STOPAFT, and FILSZ, can contain up to 28 digits, the last 15 of which can be significant (non-zero) digits. Values specified for LOCALE can contain up to 32 alphanumeric characters.
- Commas, semicolons, and blanks can be used only as delimiters. They can be used in values only if the values are constants.
- Each type of program control statement can appear only once within a single source (for example, the SYSIN data set).

**Note:** There are special rules for coding control statements in parameter lists. See *Application Programming Guide* for complete details.

# ICETOOL Control Statement Coding Rules

See *Application Programming Guide* for complete details on coding ICETOOL statements.

## Comment Statements

- A statement with an asterisk (*) in column 1 is treated as a comment statement; it is printed along with other ICETOOL statements, but not otherwise processed.
- A statement with blanks in columns 1 through 72 is treated as a blank statement; it is ignored because ICETOOL prints blank lines where appropriate.

## Operator Statements

The general format for all ICETOOL operator statements is:

```
OPERATOR operand operand ...
```

where each operand consists of KEYWORD(parameter,parameter,...) or just KEYWORD. Any number of operators can be specified in any order.

The following rules apply:

- The operator and operands must be in uppercase EBCDIC.
- The operator must be specified first.
- One blank is required between the operator and the first operand.
- One blank is required between operands.
- Any number of blanks can be specified before or after the operator or any operand, but blanks cannot be specified anywhere else except within quoted character strings.
- Parentheses must be used where shown. Commas or semicolons must be used where commas are shown.
- Operands can be in any order.
- Columns 1 through 72 are scanned. Columns 73 through 80 are ignored.
- Continuation can be indicated by a hyphen (-) *after* the operator or any operand. The next operand must then be specified on the next line. For example:

```
SORT FROM(INDD) -
   USING(ABCD) -
   TO(OUTPUT1,OUTPUT2,OUTPUT3)
```

  Any characters specified after the hyphen are ignored. Each operand *must* be completely specified on one line.

## SYMNAMES Statement Coding Rules

See *Application Programming Guide* for complete details on coding SYMNAMES statements.

## Comment Statements

- A statement with an asterisk (*) in column 1 is treated as a comment statement; it is printed in SYMNOUT (if specified), but not otherwise processed.
- A statement with blanks in columns 1 through 80 is treated as a blank statement; it is printed in SYMNOUT (if specified), but not otherwise processed.

## Symbol Statements

The general format for a symbol statement is:

```
symbol,value remark
```

The general coding rules are as follows:

- Columns 1 through 80 are scanned.
- The symbol can start in column 1 or in any column after 1.
- A remark is optional, but if specified, must be separated from the value by at least one blank. A remark is printed in SYMNOUT (if specified), but otherwise not processed.
- A semicolon(;) can be used instead of a comma (,) between the symbol and the value.
- Continuation is not allowed. Each symbol and value must be completely specified on one line.

## Keyword Statements

The general format for a keyword statement is:

```
keyword,value remark
```

The general coding rules are as follows:

- Columns 1 through 80 are scanned.
- The keyword can start in column 1 or in any column after 1.
- The keyword must be specified in all **uppercase** letters. Otherwise, it will be treated as a symbol.
- A remark is optional, but if specified, must be separated from the value by at least one blank. A remark is printed in SYMNOUT (if specified), but otherwise not processed.
- A semicolon(;) can be used instead of a comma (,) between the keyword and the value.
- Continuation is not allowed. Each keyword and value must be completely specified on one line.

## SYMNAMES Statement Coding Rules

# Chapter 2. DFSORT Program Control Statements

## ALTSEQ–Altering EBCDIC Collating Sequence

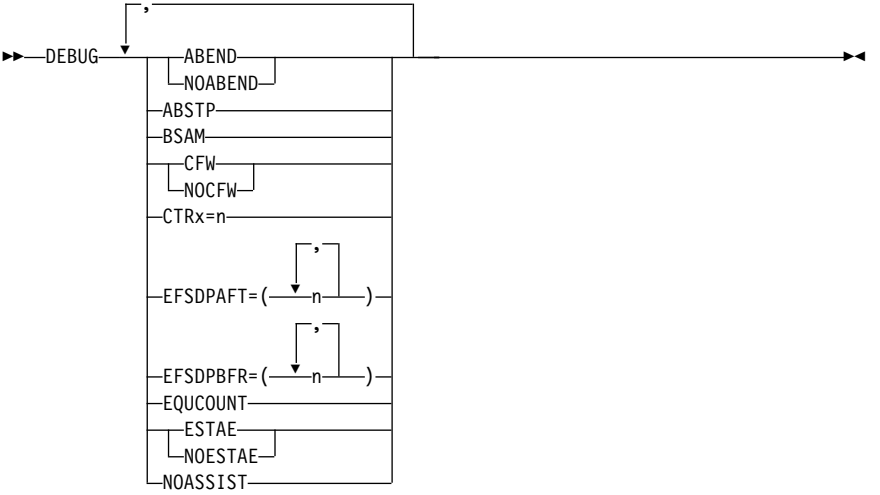

**Example**

```
ALTSEQ CODE=(5BEA)
SORT FIELDS=(18,20,AQ,A)
```

The character $ (X'5B') is to collate at position X'EA', that is, after uppercase Z (X'E9').

| Operand | Description |
|---|---|
| CODE= | Specifies changes to the alternate translation table (ALTSEQ table). Any modifications you specify are applied to the standard EBCDIC translation table. The ALTSEQ table is used to apply an alternate collating sequence for AQ fields (or CH fields with CHALT in effect), and to convert characters for fields with TRAN=ALTSEQ.<br>**ff** specifies, in hexadecimal, the character whose position is to be changed in the ALTSEQ table.<br>**tt** specifies, in hexadecimal, the new position the character is to occupy in the ALTSEQ table. |

## DEBUG—Specifying Diagnostic Options




**Example**

```
DEBUG BSAM
```

Use BSAM access method instead of EXCP for the input and output data sets.

| Operand | Description |
|---|---|
| ABEND | Terminates an unsuccessful run with a user abend. |
| NOABEND | Terminates an unsuccessful run with a return code of 16. |
| ABSTP | Terminates an unsuccessful run with an abend. Prevents loss of needed information in a dump when NOESTAE is in effect. |
| BSAM | Bypasses EXCP access method for input and output data sets. |
| CFW | Specifies that DFSORT can use cache fast write when processing SORTWKdd data sets. |
| NOCFW | Specifies that DFSORT cannot use cache fast write. |

| Operand | Description |
|---|---|
| CTRx= | Abends when the input or output record count equals n.<br><br>Valid values for x are:<br>**2** Count of input records being moved from the input buffer<br>**3** Count of output records being moved to the output buffer<br>**4** Count of input records inserted by E15<br>**5** Count of output records deleted by E35 |
| EFSDPAFT= | Causes a SNAP dump after a major call to an EFS program. You can specify up to 4 options in any combination.<br><br>Valid values are:<br>**2** Take the SNAP dump after Major Call 2 to the EFS program.<br>**3** Take the SNAP dump after Major Call 3 to the EFS program.<br>**4** Take the SNAP dump after Major Call 4 to the EFS program.<br>**5** Take the SNAP dump after Major Call 5 to the EFS program. |
| EFSDPBFR= | Causes a SNAP dump before a major call to an EFS program. You can specify up to 4 options in any combination.<br><br>Valid values are:<br>**2** Take the SNAP dump before Major Call 2 to the EFS program.<br>**3** Take the SNAP dump before Major Call 3 to the EFS program.<br>**4** Take the SNAP dump before Major Call 4 to the EFS program.<br>**5** Take the SNAP dump before Major Call 5 to the EFS program. |
| EQUCOUNT | Counts the number of sorted records with equal (duplicate) keys and prints the amount in message ICE184I. |
| ESTAE | Specifies that DFSORT can use its ESTAE recovery routine during the entire run. |
| NOESTAE | Directs DFSORT to delete the ESTAE recovery routine. If an abend occurs when NOESTAE is in effect, DFSORT abend recovery processing will not occur. |
| NOASSIST | Indicates not to use System/370™-XA Sorting Instructions. |

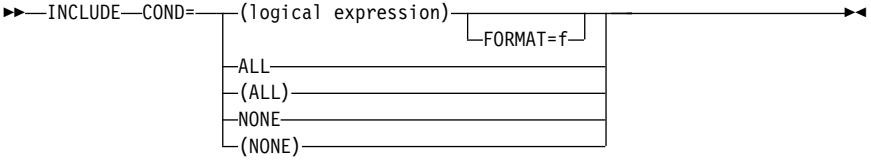
## END—Discontinuing Reading Control Statements


**Example**

```
END
```

Causes DFSORT to discontinue reading control statements in SYSIN, DFSPARM, or SORTCNTL. Must precede object decks in SYSIN.

## INCLUDE—Including Records in the Output Data Set



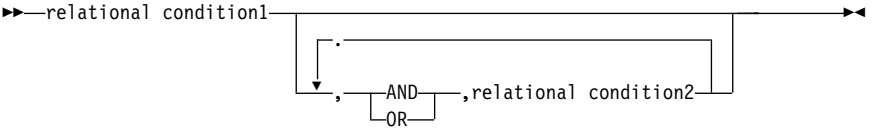

INCLUDE and OMIT control statements are mutually exclusive.

**Example**

```
INCLUDE COND=(1,10,CH,EQ,50,10,CH,OR,
              25,1,ALL,X'4C')
```

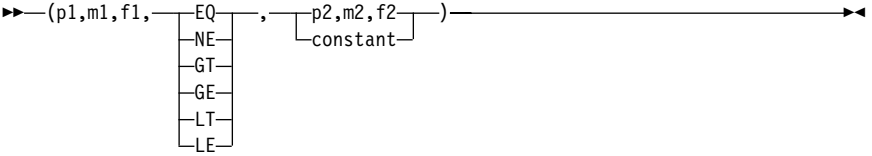
Include in the output data set only those records in which the 10-byte character field, starting at byte 1, is equal to the 10-byte character field, starting at byte 50, or in which byte 25 has all of bits 1, 4, and 5 on.

Logical expression is one or more relational conditions:

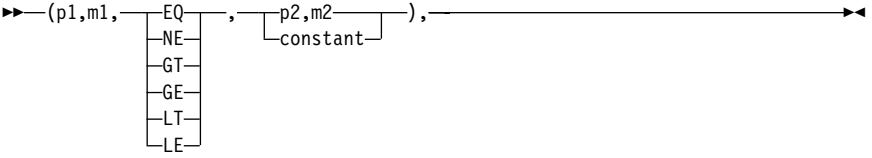



Relational conditions are of four types:

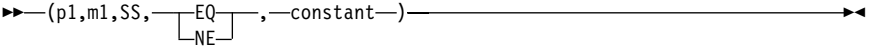
## 1) Comparison




Or, if FORMAT=f is used:



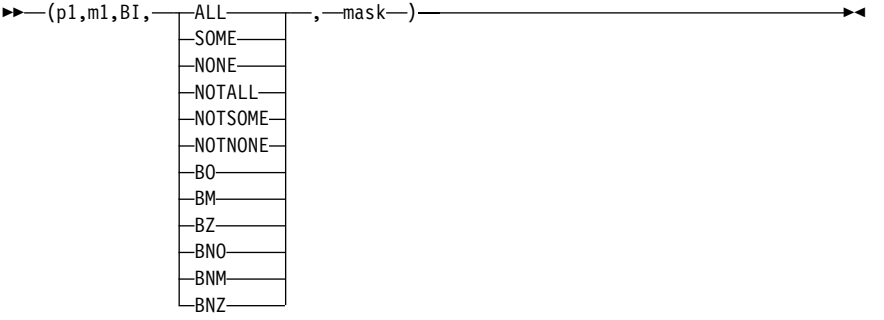

## 2) Substring comparison



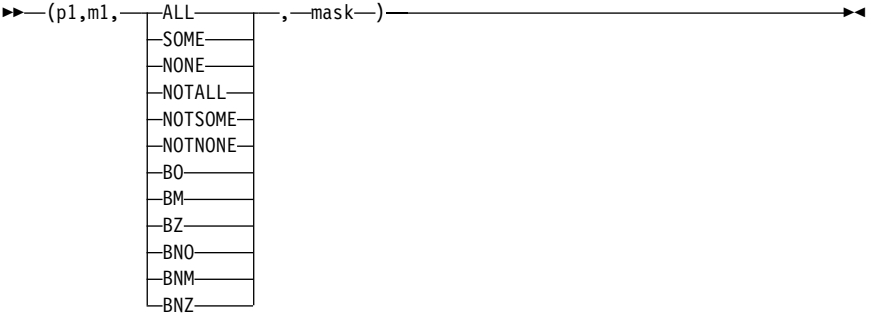

Or, if FORMAT=SS is used:

## 3) Bit Logic

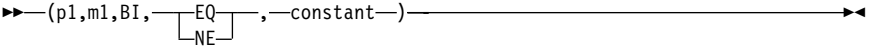
### Method 1: Bit Operator




Or, if FORMAT=BI is used:

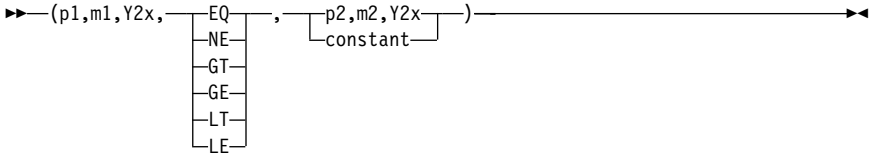



### Method 2: Bit Comparison

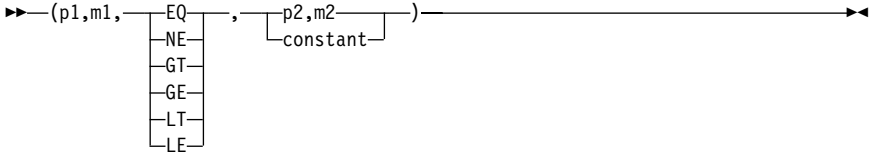



Or, if FORMAT=BI is used:

## 4) Date Comparison




Or, if FORMAT=Y2x is used:



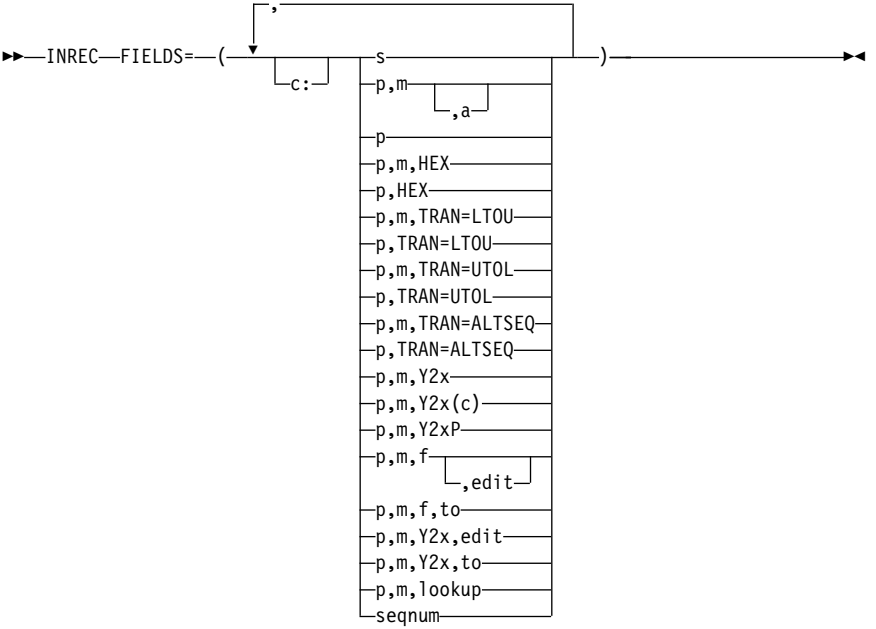
## INCLUDE—Including Records in the Output Data Set

| Operand | Description |
|---|---|
| COND= | Describes relational conditions.<br><br>**logical expression**<br>One or more relational conditions.<br>**p** Position of input field; a byte location within a record that is the first byte of an input field.<br>**m** Length of input field.<br>**f** Format of input field.<br>**SS** Substring format.<br>**Y2x** Date format.<br>**EQ** Equal to.<br>**NE** Not equal to.<br>**GT** Greater than.<br>**GE** Greater than or equal to.<br>**LT** Less than.<br>**LE** Less than or equal to.<br>**ALL or BO**<br>All mask bits are on.<br>**SOME or BM**<br>Some but not all mask bits are on.<br>**NONE or BZ**<br>No mask bits are on.<br>**NOTALL or BNO**<br>Some or no mask bits are on.<br>**NOTSOME or BNM**<br>All or no mask bits are on.<br>**NOTNONE or BNZ**<br>All or some mask bits are on.<br>**constant** Can be decimal (n, +n, -n, DATEnP), character (C'string', DATEn, DATEn(c)), hexadecimal (X'value') or two-digit year date (Y'value').<br>**mask** Can be hexadecimal (X'value') or bit (B'value').<br>**AND** Logical AND.<br>**OR** Logical OR.<br><br>**ALL (ALL)**<br>Include all input records.<br><br>**NONE (NONE)**<br>Include no input records. |
| FORMAT= | Can be used only when all input fields in the entire logical expression have the same format. |

## Compare Field Formats for INCLUDE

| Format | Length | Description |
|---|---|---|
| CH | 1 to 4092 bytes | Character (if CHALT is in effect, CH is treated as AQ) |
| AQ | 1 to 256 bytes | Character, alternate collating sequence. |
| ZD | 1 to 256 bytes | Signed zoned decimal |
| PD | 1 to 255 bytes | Signed packed decimal |
| FI | 1 to 256 bytes | Signed fixed point |
| BI | 1 to 256 bytes | Unsigned binary |
| AC | 1 to 256 bytes | ISCII/ASCII character |
| CSF or FS | 1 to 16 bytes | Signed numeric, optional leading floating sign. |
| CSL or LS | 2 to 256 bytes | Signed numeric, leading separate sign |
| CST or TS | 2 to 256 bytes | Signed numeric, trailing separate sign |
| CLO or OL | 1 to 256 bytes | Signed numeric, leading overpunch sign |
| CTO or OT | 1 to 256 bytes | Signed numeric, trailing overpunch sign |
| ASL | 2 to 256 bytes | Signed ISCII/ASCII numeric, leading separate sign |
| AST | 2 to 256 bytes | Signed ISCII/ASCII numeric, trailing separate sign |
| D2 | 1 to 256 bytes | User-defined data type (requires an EFS program) |
| Y2x | 1 to 6 bytes | Date format |

## INREC—Reformatting Records Before Processing




**Example**

```
INREC FIELDS=(20,4,12,3)
```

Reformats input records, positions 20 through 23 and positions 12 through 14, into positions 1 through 4 and 5 through 7, respectively.

### FIELDS=

Order of separation fields, unedited and edited input fields, and sequence numbers in the reformatted input record.

| Operand | Description |
|---|---|
| c: | Column for separation field, input field or sequence number in the reformatted input record. |

| Operand | Description |
|---|---|
| s | A separation field appears in the reformatted input record. Permissible values: nX (blank separation); nZ (binary zero separation); nC'xx...x' (character string separation); nX'yy...yy' (hexadecimal string separation); DATEnP (packed decimal current date); DATEn and DATEn(c) (character current date); TIMEnP (packed decimal current time); TIMEn and TIMEn(c) (character current time). |
| p,m,a | Unedited input field.<br>**p** Position of input field; a byte location within a record that is the first byte of an input field.<br>**m** Length of input field.<br>**a** Alignment of input field in the reformatted input record. Permissible values: H (halfword aligned); F (fullword aligned); D (doubleword aligned). |
| p | Unedited variable part of input record.<br>See p under p,m,a above. |
| p,m,HEX | Hexadecimal representation of input field.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**HEX** Requests hexadecimal representation of input field. |
| p,HEX | Hexadecimal representation of variable part of input record.<br>**p** See p under p,m,a above.<br>**HEX** Requests hexadecimal representation of variable part of input record. |
| p,m,TRAN=LTOU | Uppercase representation of input field.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**TRAN=LTOU** Requests lowercase (a-z) to uppercase (A-Z) conversion. |
| p,TRAN=LTOU | Uppercase representation of variable part of input record.<br>**p** See p under p,m,a above.<br>**TRAN=LTOU** Requests lowercase (a-z) to uppercase (A-Z) conversion. |

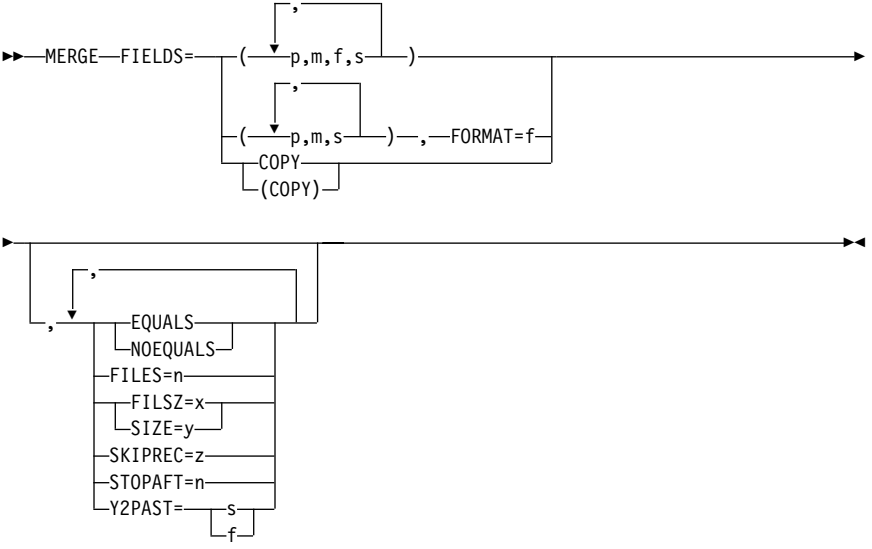
| Operand | Description |
|---|---|
| p,m,TRAN=UTOL | Lowercase representation of input field.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**TRAN=UTOL**<br>Requests uppercase (A-Z) to lowercase (a-z) conversion. |
| p,TRAN=UTOL | Lowercase representation of variable part of input record.<br>**p** See p under p,m,a above.<br>**TRAN=UTOL**<br>Requests uppercase (A-Z) to lowercase (a-z) conversion. |
| p,m,TRAN=ALTSEQ | Representation of input field according to the ALTSEQ table.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**TRAN=ALTSEQ**<br>Requests conversion of characters according to the ALTSEQ table. |
| p,TRAN=ALTSEQ | Representation of variable part of input record according to the ALTSEQ table.<br>**p** See p under p,m,a above.<br>**TRAN=ALTSEQ**<br>Requests conversion of characters according to the ALTSEQ table. |
| p,m,Y2x | Four-digit year CH date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** Date format. |
| p,m,Y2x(c) | Four-digit year CH date representation with separators of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** Date format.<br>**c** Separator character. |

| Operand | Description |
|---|---|
| p,m,Y2xP | Four-digit year PD date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2xP** Date format. |
| p,m,f,edit | Edited numeric input field.<br>**p** See p under p,m,a above.<br>**m** Length of numeric field.<br>**f** Format of numeric field: BI, FI, ZD, PD, PD0, FS, CSF, DTn, or TMn.<br>**edit** How field is edited.<br>See *Application Programming Guide* for details. |
| p,m,f,to | Converted numeric input field.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,f,edit above.<br>**f** See f under p,m,f,edit above.<br>**to** How field is converted.<br>See *Application Programming Guide* for details. |
| p,m,Y2x,edit | Edited four-digit year CH date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** See Y2x under p,m,Y2x above.<br>**edit** See edit under p,m,f,edit above. |
| p,m,Y2x,to | Converted four-digit year date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** See Y2x under p,m,Y2x above.<br>**to** See to under p,m,f,to above. |

## INREC—Reformatting Records Before Processing

| Operand | Description |
|---|---|
| p,m,lookup | Character or hexadecimal string from lookup table.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**lookup** How input field is changed using a lookup table.<br>See *Application Programming Guide* for details. |
| seqnum | Sequence number: BI, PD, ZD, FS or CSF.<br>See *Application Programming Guide* for details. |

## MERGE—Specifying a MERGE or COPY




SORT and MERGE control statements are mutually exclusive.

**Example**

```
MERGE FIELDS=(2,5,CH,A)
```

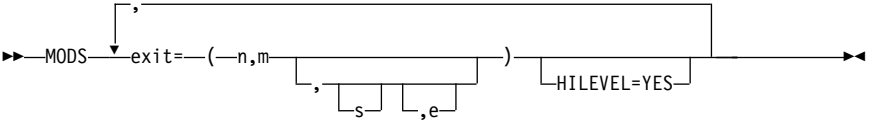
The input record control field starts at byte 2, is 5 bytes long, and contains character (EBCDIC) data in ascending order.

| Operand | Description |
|---|---|
| FIELDS= | Control fields must be described in descending order of significance.<br>**p** Position of control field; a byte location within a record that is the first byte of a control field.<br>**m** Length of control field.<br>**f** Format of control field.<br>**s** Sequence of data. Can be:<br>A — ascending<br>D — descending<br>E — user-modified control field (used with an E61 user exit) |
| FORMAT= | Can be used instead of the f subparameter of FIELDS= if all control field data formats are the same. |
| FIELDS=COPY<br>FIELDS=(COPY) | Copies a SORTIN data set to one or more output data sets. |
| EQUALS | Specifies that the original sequence of records that collate identically must be preserved. |
| NOEQUALS | Specifies that the original sequence of records that collate identically need not be preserved. |
| FILES | Specifies the number of input files, n, for a merge when input is supplied through an E32 user exit. |
| FILSZ= | Specifies the exact number (x) of records to be processed. |
| SIZE= | Specifies the exact number (y) of records in the input data sets. |
| SKIPREC= | Skip z input records before DFSORT begins to copy. |
| STOPAFT= | Indicates the number of records you want accepted for copying. |
| Y2PAST= | Specifies the sliding or fixed century window.<br>**s** Starts the sliding century window at the current year − s.<br>**f** Starts the fixed century window at f.<br><br>**Note:** CENTURY and CENTWIN can be used instead of Y2PAST. |

## Control Field Formats for MERGE

| Format | Length | Description |
|---|---|---|
| CH | 1 to 4092 bytes | Character (if CHALT is in effect, CH is treated as AQ) |
| AQ | 1 to 4092 bytes | Character, alternate collating sequence |
| ZD | 1 to 32 bytes | Signed zoned decimal |
| PD | 1 to 32 bytes | Signed packed decimal |
| PD0 | 2 to 8 bytes | Packed decimal with sign and first digit ignored |
| FI | 1 to 256 bytes | Signed fixed point |
| BI | 1 bit to 4092 bytes | Unsigned binary |
| FL | 1 to 256 bytes | Signed floating point |
| AC | 1 to 4092 bytes | ISCII/ASCII character |
| CSF or FS | 1 to 16 bytes | Signed numeric, optional leading floating sign. |
| CSL or LS | 2 to 256 bytes | Signed numeric, leading separate sign |
| CST or TS | 2 to 256 bytes | Signed numeric, trailing separate sign |
| CLO or OL | 1 to 256 bytes | Signed numeric, leading overpunch sign |
| CTO or OT | 1 to 256 bytes | Signed numeric, trailing overpunch sign |
| ASL | 2 to 256 bytes | Signed ISCII/ASCII numeric, leading separate sign |
| AST | 2 to 256 bytes | Signed ISCII/ASCII numeric, trailing separate sign |
| D1 | 1 to 4092 bytes | User-defined data type (requires an EFS program) |
| Y2x | 1 to 6 bytes | Date format |

## MODS—Identifying User Exit Routines




**Example**

```
MODS E15=(ADDREC,552,MODLIB),
         E35=(ALTREC,11032,,C)
```

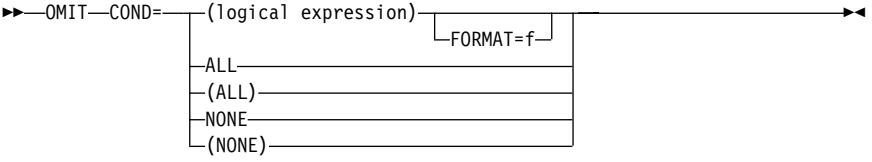
At user exit E15, DFSORT transfers control to your user exit routine. Your user exit routine is in the library defined by the job control statement with the ddname MODLIB. Its member name is ADDREC and it uses 552 bytes.

At user exit E35, DFSORT transfers control to your COBOL user exit routine. Your user exit routine is in the STEPLIB/JOBLIB or link libraries. Its member name is ALTREC and it uses 11032 bytes.

## MODS—Identifying User Exit Routines

| Operand | Description |
|---|---|
| exit= | Specifies a user exit at which DFSORT passes control to a user exit routine; for example, E15 or E35.<br>**n** Name of user exit routine or member name, if the user exit routine is in a library.<br>**m** Number of bytes of main storage the user exit routine uses.<br>**s** Name of DD statement that defines the library in which your user exit routine is located or name of SYSIN if your user exit routine is in the input stream. Do not specify s if your user exit routine is located in the STEPLIB/JOBLIB or link libraries.<br>**e** Indicates the linkage-editor requirements of your user exit routine, or indicates your user exit routine is written in COBOL.<br>**N** Specifies that your user exit routine has already been link-edited and can be used in the DFSORT run without further link-editing. This is the default.<br>**C** Specifies that your E15 or E35 user exit routine is written in COBOL.<br>**T** Specifies that your user exit routine must be link-edited together with other user exit routines.<br>**S** Specifies that your E11 or E31 user exit routine must be link-edited separately from other user exit routines. |
| HILEVEL=YES | Specifies that your E15 routine, if specified, and E35 routine, if specified, are written in COBOL.<br>**Note:** COBOL=YES can be used instead of HILEVEL=YES. |

## OMIT—Omitting Records from the Output Data Set




INCLUDE and OMIT control statements are mutually exclusive.

See INCLUDE control statement for more detailed syntax, descriptions of operands, and compare field formats.

**Example**

```
OMIT COND=(1,2,CH,EQ,C'T1',AND,
          25,1,EQ,B'.10.....')
```

Omit from the output data set those records in which the 2-byte character field, starting at byte 1, is equal to the characters 'T1', and in which byte 25 has bit 1 on and bit 2 off.

## OPTION—Specifying DFSORT Options or COPY

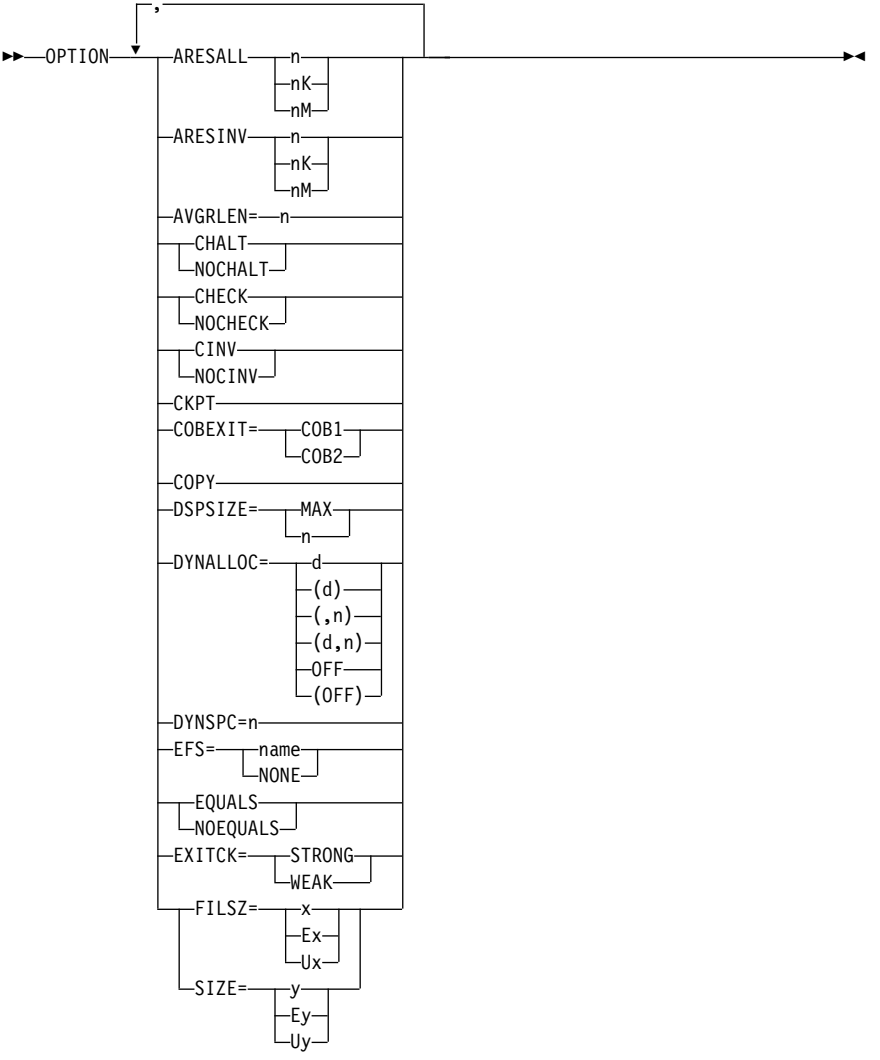



*Figure 1. Syntax Diagram for the Option Control Statement (Part 1 of 3)*

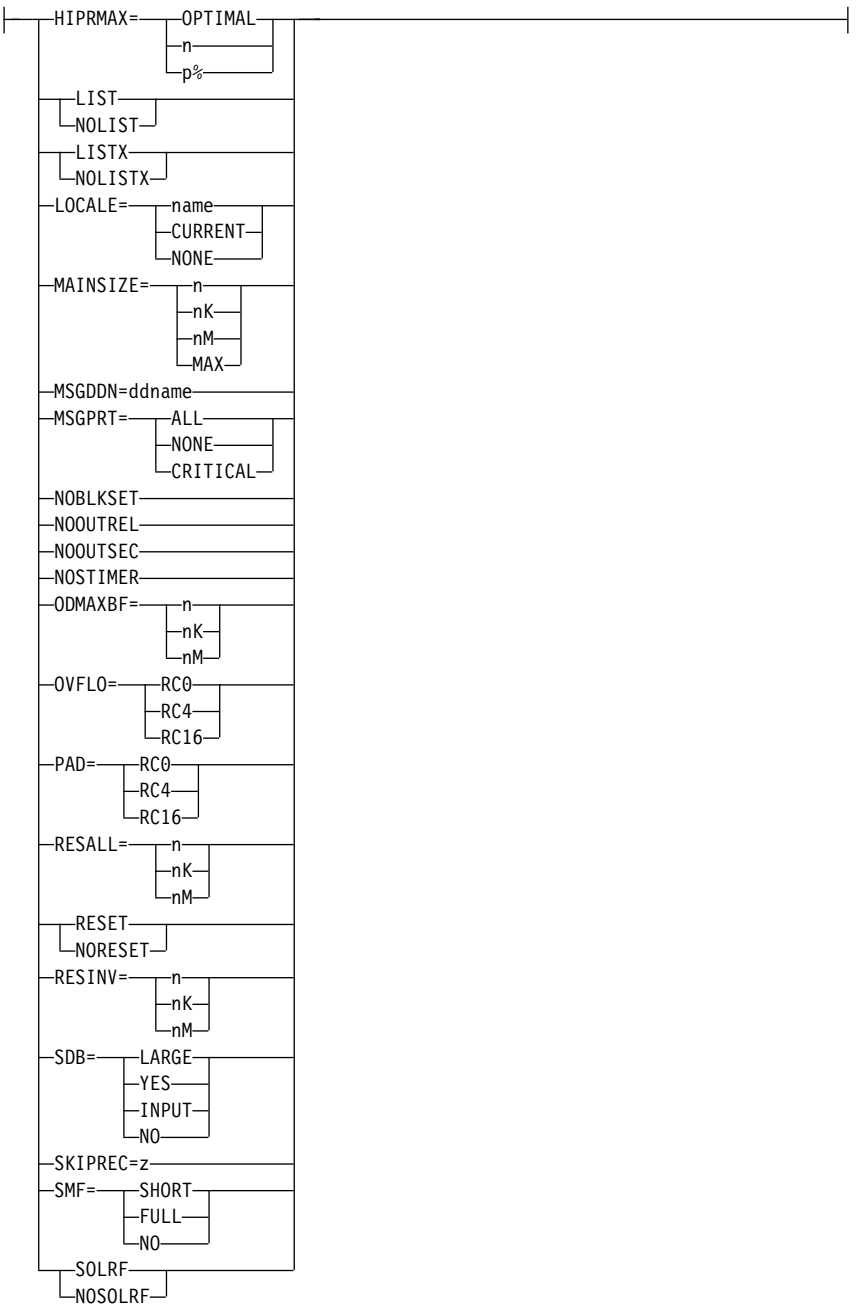



*Figure 1. Syntax Diagram for the Option Control Statement (Part 2 of 3)*

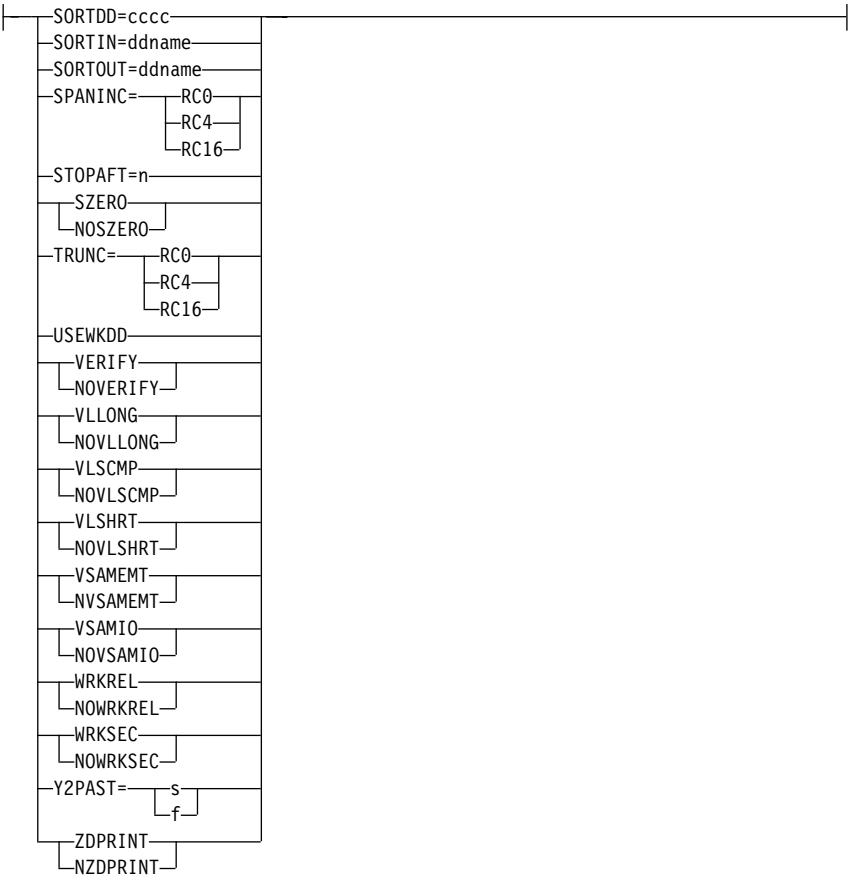



*Figure 1. Syntax Diagram for the Option Control Statement (Part 3 of 3)*

**Example**

```
OPTION DYNALLOC=(3390,6),SKIPREC=5
```

Dynamically allocates up to six work data sets on 3390 DASD, and skips five records at the beginning of the input data set.

| Operand | Description |
|---|---|
| ARESALL= | Reserves main storage above 16-megabyte virtual for system and application use.<br>**n** Reserves n bytes of main storage.<br>**nK** Reserves n times K (K=1024) bytes of main storage.<br>**nM** Reserves n times M (M=1 048 576) bytes of main storage. |
| ARESINV= | Reserves main storage above 16-megabyte virtual when DFSORT is dynamically invoked.<br>**n** Reserves n bytes of main storage.<br>**nK** Reserves n times K (K=1024) bytes of main storage.<br>**nM** Reserves n times M (M=1 048 576) bytes of main storage. |
| AVGRLEN= | Specifies the average input record length in bytes for variable-length record sort applications.<br>**Note:** L5=n can be used instead of AVGRLEN=n. |
| CHALT | Translates both AQ and CH format control fields using the alternate collating sequence. |
| NOCHALT | Translates only AQ format control fields. |
| CHECK | Checks record counters at the end of program execution. |
| NOCHECK | Suppresses checking record counters when an E35 user exit routine is used without an output data set. |
| CINV | Allows control interval access for VSAM data sets. |
| NOCINV | Does not allow control interval access for VSAM data sets. |
| CKPT | Activates the operating system's checkpoint/restart facility. |
| COBEXIT= | Specifies the library for COBOL E15 and E35 routines.<br>**COB1** COBOL E15 and E35 routines run with the OS/VS COBOL run-time library or with no COBOL run-time library.<br>**COB2** COBOL E15 and E35 routines run with either the VS COBOL II run-time library or the Language Environment run-time library. |
| COPY | Copies a SORTIN data set to one or more output data sets. |

| Operand | Description |
|---|---|
| DSPSIZE= | Specifies the maximum amount of data space to be used with dataspace sorting.<br>**MAX** Specifies that DFSORT dynamically determines the maximum amount of data space to be used.<br>**n** Specifies the maximum amount, in megabytes, of data space to be used. |
| DYNALLOC | Allocates work data sets dynamically with values for d and n specified (or defaulted) by the DYNALOC option of the ICEMAC installation macro. |
| DYNALLOC= | Allocates work data sets dynamically.<br>**d** Specifies the device name.<br>**n** Specifies the maximum number of requested work data sets. |
| DYNALLOC=OFF | Prevents dynamic allocation of work data sets. |
| DYNSPC= | Specifies the total primary space for all of the dynamically allocated work data sets when the file size is unknown.<br>**n** Specifies the primary space in megabytes. |
| EFS= | Specifies the name of an EFS program for DFSORT to call.<br>**name** Calls the EFS program whose name you specify.<br>**NONE** Does not call an EFS program. |
| EQUALS | Specifies that the original sequence of records that collate identically must be preserved. |
| NOEQUALS | Specifies that the original sequence of records that collate identically need not be preserved. |
| EXITCK= | Specifies whether DFSORT terminates or continues processing when an E15 or E35 exit passes an invalid return code.<br>**STRONG** Terminates for invalid return code from E15/E35.<br>**WEAK** Continues processing for invalid return code from E15/E35. |
| FILSZ= | Specifies the number of records to be processed.<br>**x** The exact number of records to be processed.<br>**Ex** The estimated number of records to be processed. x must be immediately preceded by the letter E.<br>**Ux** The number of records to be processed. x must be immediately preceded by the letter U. |

| Operand | Description |
|---|---|
| SIZE= | Specifies the number of records in the input data sets.<br>**y** The exact number of records in the input data sets.<br>**Ey** The estimated number of records in the input data sets. y must be immediately preceded by the letter E.<br>**Uy** The number of records in the input data sets. y must be immediately preceded by the letter U. |
| HIPRMAX= | Allocates Hiperspace™ for Hipersorting™.<br>**OPTIMAL** Determines dynamically the maximum amount of Hiperspace.<br>**n** Determines dynamically the maximum amount of Hiperspace up to n megabytes.<br>**p%** Determines dynamically the maximum amount of Hiperspace up to p percent of configured expanded storage (31-bit mode) or of a portion of central storage (64-bit mode).<br>**Note:** HIPRLIM=OPTIMAL can be used instead of HIPRMAX=OPTIMAL. HIPRLIM=n (n times 4096 bytes) can be used instead of HIPRMAX=n (megabytes). |
| LIST | Prints DFSORT control statements. |
| NOLIST | Does not print DFSORT control statements. |
| LISTX | Prints control statements returned by an EFS program. |
| NOLISTX | Does not print control statements returned by an EFS program. |
| LOCALE= | Designates active locale.<br>**name** Name of active locale.<br>**CURRENT** Current locale remains active.<br>**NONE** Does not use locale processing. |

## OPTION—Specifying DFSORT Options or COPY

| Operand | Description |
|---|---|
| MAINSIZE= | Specifies the amount of main storage available:<br>**n** Allocates n bytes of main storage.<br>**nK** Allocates n times K (K=1024) bytes of main storage.<br>**nM** Allocates n times M (M=1 048 576) bytes of main storage.<br>**MAX** Allocates main storage specified by TMAXLIM or MAXLIM installation option.<br>**Note:** CORE can be used instead of MAINSIZE. |
| MSGDDN= | Specifies the message data set ddname. |
| MSGPRT= | Specifies the message type.<br>**ALL** Prints all messages, except diagnostic messages.<br>**NONE** Does not print messages and control statements.<br>**CRITICAL** Prints only critical (error) messages.<br>**Note:** PRINT can be used instead of MSGPRT. |
| NOBLKSET | Bypasses the Blockset technique. |
| NOOUTREL | Does not release unused temporary output data set space. |
| NOOUTSEC | Does not use automatic secondary allocation for temporary or new output data sets. |
| NOSTIMER | Does not use the STIMER macro. |
| ODMAXBF= | Specifies maximum amount of OUTFIL data set buffer space.<br>**n** Specifies n bytes of buffer space.<br>**nK** Specifies n times K (K=1024) bytes of buffer space.<br>**nM** Specifies n times M (M=1 048 576) bytes of buffer space. |
| OVFLO= | Specifies the action to be taken by DFSORT when BI, FI, PD or ZD summary fields overflow.<br>**RC0** Continues processing and sets a return code of 0.<br>**RC4** Continues processing and sets a return code of 4.<br>**RC16** Terminates and gives a return code of 16. |

| Operand | Description |
|---|---|
| PAD= | Specifies the action to be taken by DFSORT for LRECL padding.<br>**RC0** Continues processing and sets a return code of 0.<br>**RC4** Continues processing and sets a return code of 4.<br>**RC16** Terminates and gives a return code of 16. |
| RESALL= | Reserves main storage below 16-megabyte virtual for system and application use when SIZE/MAINSIZE=MAX is in effect.<br>**n** Reserves n bytes of main storage.<br>**nK** Reserves n times K (K=1024) bytes of main storage.<br>**nM** Reserves n times M (M=1 048 576) bytes of main storage. |
| RESET | Processes a VSAM output data set with REUSE as NEW. |
| NORESET | Processes a VSAM output data set with REUSE as MOD. |
| RESINV= | Reserves main storage below 16-megabyte virtual for an invoking program, when SIZE/MAINSIZE=MAX is in effect.<br>**n** Reserves n bytes of main storage.<br>**nK** Reserves n times K (K=1024) bytes of main storage.<br>**nM** Reserves n times M (M=1 048 576) bytes of main storage. |

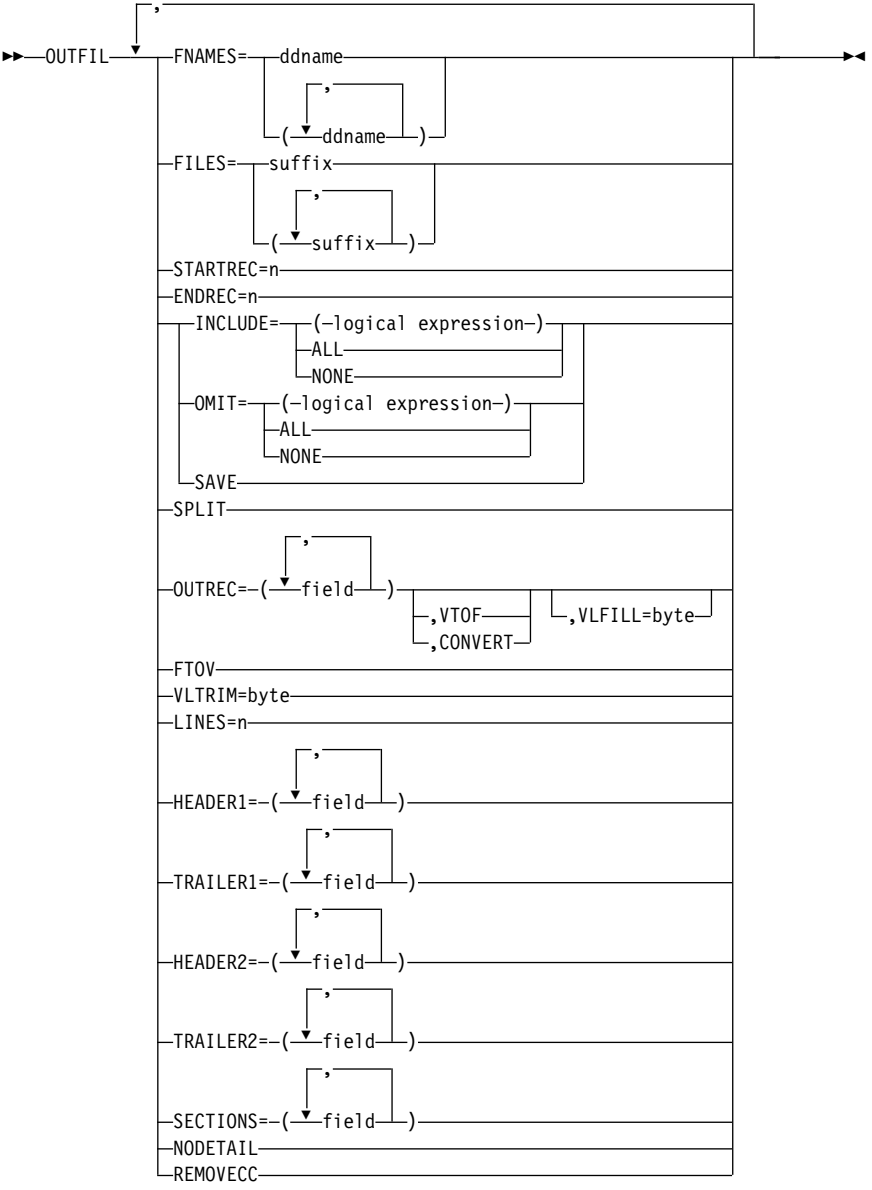
| Operand | Description |
|---|---|
| SDB= | Specifies whether DFSORT should use the system-determined optimum block size for output data sets when the block size is specified as zero or defaulted to zero.<br>**LARGE** Uses the system-determined optimum block size for output data sets. Allows DFSORT to select a block size greater than 32760 bytes for a tape output data set, when appropriate.<br>**YES** Uses the system-determined optimum block size for output data sets, but limits the selected block size to a maximum of 32760 bytes.<br>**Note:** SDB and SDB=SMALL can be used instead of SDB=YES.<br>**INPUT** Uses the system-determined optimum block size for output data sets, but limits the selected block size to a maximum of 32760 bytes if the input block size is less than or equal to 32760 bytes.<br>**NO** Does not use the system-determined optimum block size for output data sets.<br>**Note:** NOSDB can be used instead of SDB=NO. |
| SKIPREC= | Skips z input records before DFSORT begins to sort or copy. |
| SMF= | Indicates whether an SMF type-16 record will be produced.<br>**SHORT** Produces a short SMF record.<br>**FULL** Produces a full SMF record.<br>**NO** Does not produce an SMF record. |
| SOLRF | Uses the reformatted record length for the SORTOUT LRECL. |
| NOSOLRF | Does not use the reformatted record length for the SORTOUT LRECL. |
| SORTDD= | Specifies the prefix for certain “SORT ” ddnames. |
| SORTIN= | Specifies the ddname to be associated with the SORTIN data set. |
| SORTOUT= | Specifies the ddname to be associated with the SORTOUT data set. |

| Operand | Description |
|---|---|
| SPANINC= | Specifies the action to be taken by DFSORT when incomplete spanned records are detected.<br>**RC0** Continues processing and sets a return code of 0.<br>**RC4** Continues processing and sets a return code of 4.<br>**RC16** Terminates and gives a return code of 16. |
| STOPAFT= | Indicates the number of records you want accepted for sorting or copying. |
| SZERO | Treats − 0 and + 0 as signed, that is, as the same. |
| NOSZERO | Treats − 0 and + 0 as unsigned, that is, as different. |
| TRUNC= | Specifies the action to be taken by DFSORT for LRECL truncation.<br>**RC0** Continues processing and sets a return code of 0.<br>**RC4** Continues processing and sets a return code of 4.<br>**RC16** Terminates and gives a return code of 16. |
| USEWKDD | Uses JCL SORTWKdd data sets, if present. |
| VERIFY | Verifies the sequence of the final output records.<br>**Note:** SEQ=YES can be used instead of VERIFY. |
| NOVERIFY | Does not verify the sequence of the final output records.<br>**Note:** SEQ=NO can be used instead of NOVERIFY. |
| VLLONG | Truncates long variable-length output records. |
| NOVLLONG | Stops processing if a long variable-length output record is found. |
| VLSCMP | Pads short variable-length compare fields with binary zeros. |
| NOVLSCMP | Does not pad short variable-length compare fields. |
| VLSHRT | Continues processing if a short variable-length control field, compare field, or summary field is found. |
| NOVLSHRT | Stops processing if a short variable-length control field, compare field, or summary field is found. |
| VSAMEMT | Processes an empty VSAM input data set. |
| NVSAMEMT | Terminates for an empty VSAM input data set. |
| VSAMIO | Allows same VSAM data set with REUSE to be sorted in-place. |
| NOVSAMIO | Terminates if same VSAM data set used for input and output. |

## OPTION—Specifying DFSORT Options or COPY

| Operand | Description |
|---|---|
| WRKREL | Releases unused temporary SORTWKdd data set space.<br>**Note:** RLS=n (n greater than 0) can be used instead of WRKREL. |
| NOWRKREL | Does not release unused temporary SORTWKdd data set space.<br>**Note:** RLS=0 can be used instead of NOWRKREL. |
| WRKSEC | Uses automatic secondary allocation for temporary SORTWKdd data sets. 25% of the primary allocation is used.<br>**Note:** SEC=n (n greater than 0) can be used instead of WRKSEC. |
| NOWRKSEC | Does not use automatic secondary allocation for temporary SORTWKdd data sets.<br>**Note:** SEC=0 can be used instead of NOWRKSEC. |
| Y2PAST= | Specifies the sliding or fixed century window.<br>**s** Starts the sliding century window at the current year − s.<br>**f** Starts the fixed century window at f.<br>**Note:** CENTURY and CENTWIN can be used instead of Y2PAST. |
| ZDPRINT | Converts positive ZD summation results to printable numbers.<br>**Note:** ZDPRINT=YES can be used instead of ZDPRINT. |
| NZDPRINT | Does not convert positive ZD summation results to printable numbers.<br>**Note:** ZDPRINT=NO can be used instead of NZDPRINT. |

## OUTFIL—Creating Multiple Output Data Sets



**Example**

```
OUTFIL FNAMES=(ACTG,BKUPACTG),
   INCLUDE=(8,15,CH,EQ,C'ACCOUNTING')
OUTFIL FNAMES=(DEV,BKUPDEV),
   INCLUDE=(8,15,CH,EQ,C'DEVELOPMENT')
OUTFIL FNAMES=OTHER,SAVE
```

With a single pass over the input data set:

- Include in the ACTG and BKUPACTG output data sets only those records that contain 'ACCOUNTING ' in the 15-byte character field starting at byte 8
- Include in the DEV and BKUPDEV output data sets only those records that contain 'DEVELOPMENT ' in the 15-byte character field starting at byte 8
- Include in the OTHER output data set all records not included in either the ACTG or DEV data sets.

## FNAMES=

Specifies ddnames of OUTFIL data sets.

| Operand | Description |
|---|---|
| ddname | A ddname for an OUTFIL data set. |

## FILES=

Specifies suffixes for ddnames of OUTFIL data sets.

| Operand | Description |
|---|---|
| d | A ddname of SORTOFd for an OUTFIL data set. |
| dd | A ddname of SORTOFdd for an OUTFIL data set. |
| OUT | A ddname of SORTOUT for an OUTFIL data set. |

## STARTREC=

Specifies starting record.

| Operand | Description |
|---|---|
| n | Relative record number. |

## ENDREC=

Specifies ending record.

| Operand | Description |
|---|---|
| n | Relative record number. |

## INCLUDE=

Selects records to include.

| Operand | Description |
|---|---|
| logical expression | One or more relational conditions. |
| ALL (ALL) | Includes all records. |
| NONE (NONE) | Includes no records. |

## OMIT=

Selects records to omit.

| Operand | Description |
|---|---|
| logical expression | One or more relational conditions. |
| ALL (ALL) | Omits all records. |
| NONE (NONE) | Omits no records. |

## SAVE

Selects records not otherwise included.

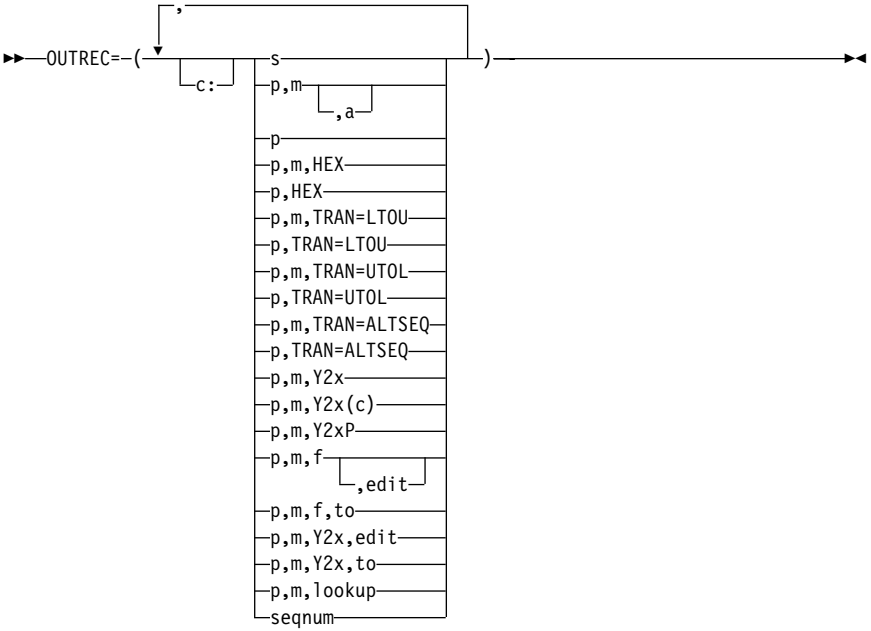
## SPLIT

Splits records in rotation.

## OUTREC=

Order of separation fields, unedited and edited input fields, and sequence numbers in the reformatted output record.

## OUTFIL—Creating Multiple Output Data Sets




| Operand | Description |
|---|---|
| c: | Column for separation field, input field or sequence number in the reformatted output record. |
| s | A separation field appears in the reformatted output record or a new output record is to be started, with or without intervening blank output records. Permissible values: nX (blank separation); nZ (binary zero separation); nC'xx...x' (character string separation); nX'yy...yy' (hexadecimal string separation); DATEnP (packed decimal current date); DATEn and DATEn(c) (character current date); TIMEnP (packed decimal current time); TIMEn and TIMEn(c) (character current time); /.../ or n/ (blank output records or new output record). |
| p,m,a | Unedited input field.<br>**p** Position of input field; a byte location within a record that is the first byte of an input field.<br>**m** Length of input field.<br>**a** Alignment of input field in the reformatted output record. Permissible values: H (halfword aligned); F (fullword aligned); D (doubleword aligned). |

| Operand | Description |
|---|---|
| p | Unedited variable part of input record.<br><br>See p under p,m,a above. |
| p,m,HEX | Hexadecimal representation of input field.<br><br>**p** See p under p,m,a above.<br><br>**m** See m under p,m,a above.<br><br>**HEX** Requests hexadecimal representation of input field. |
| p,HEX | Hexadecimal representation of variable part of input record.<br><br>**p** See p under p,m,a above.<br><br>**HEX** Requests hexadecimal representation of variable part of input record. |
| p,m,TRAN=LTOU | Uppercase representation of input field.<br><br>**p** See p under p,m,a above.<br><br>**m** See m under p,m,a above.<br><br>**TRAN=LTOU**<br>Requests lowercase (a-z) to uppercase (A-Z) conversion. |
| p,TRAN=LTOU | Uppercase representation of variable part of input record.<br><br>**p** See p under p,m,a above.<br><br>**TRAN=LTOU**<br>Requests lowercase (a-z) to uppercase (A-Z) conversion. |
| p,m,TRAN=UTOL | Lowercase representation of input field.<br><br>**p** See p under p,m,a above.<br><br>**m** See m under p,m,a above.<br><br>**TRAN=UTOL**<br>Requests uppercase (A-Z) to lowercase (a-z) converion. |
| p,TRAN=UTOL | Lowercase representation of variable part of input record.<br><br>**p** See p under p,m,a above.<br><br>**TRAN=UTOL**<br>Requests uppercase (A-Z) to lowercase (a-z) conversion. |

| Operand | Description |
|---|---|
| p,m,TRAN=ALTSEQ | Representation of input field according to the ALTSEQ table.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**TRAN=ALTSEQ** Requests conversion of characters according to the ALTSEQ table. |
| p,TRAN=ALTSEQ | Representation of variable part of input record according to the ALTSEQ table.<br>**p** See p under p,m,a above.<br>**TRAN=ALTSEQ** Requests conversion of characters according to the ALTSEQ table. |
| p,m,Y2x | Four-digit year CH date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** Date format. |
| p,m,Y2x(c) | Four-digit year CH date representation with separators of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** Date format.<br>**c** Separator character. |
| p,m,Y2xP | Four-digit year PD date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2xP** Date format. |

| Operand | Description |
|---|---|
| p,m,f,edit | Edited numeric input field.<br>**p** See p under p,m,a above.<br>**m** Length of numeric field.<br>**f** Format of numeric field: BI, FI, ZD, PD, PD0, FS, CSF, DTn, or TMn (PD0, DTn, and TMn cannot be used for TRAILERx).<br>**edit** How field is edited.<br>See *Application Programming Guide* for details. |
| p,m,f,to | Converted numeric input field.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,f,edit above.<br>**f** See f under p,m,f,edit above.<br>**to** How field is converted.<br>See *Application Programming Guide* for details. |
| p,m,Y2x,edit | Edited four-digit year CH date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** See Y2x under p,m,Y2x above.<br>**edit** See edit under p,m,f,edit above. |
| p,m,Y2x,to | Converted four-digit year date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** See Y2x under p,m,Y2x above.<br>**to** See to under p,m,f,to above. |
| p,m,lookup | Character or hexadecimal string from lookup table.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**lookup** How input field is changed using a lookup table.<br>See *Application Programming Guide* for details. |

| Operand | Description |
|---|---|
| seqnum | Sequence number: BI, PD, ZD, FS or CSF.<br><br>See *Application Programming Guide* for details. |

## VTOF or CONVERT

Specifies that variable-length input records are converted to fixed-length output records.

## VLFILL=

Allows processing of variable-length records too short to contain all OUTFIL OUTREC fields, by padding with a fill byte.

| Operand | Description |
|---|---|
| byte | Fill byte. Permissible values: C'x' (character byte); X'yy' (hexadecimal byte). |

## FTOV

Specifies that fixed-length input records are converted to variable-length output records.

## VLTRIM=

Specifies that trailing trim bytes are to be removed from the end of variable-length output records.

| Operand | Description |
|---|---|
| byte | Trim byte. Permissible values: C'x' (character byte); X'yy' (hexadecimal byte). |

## LINES=

Specifies number of lines per report page.

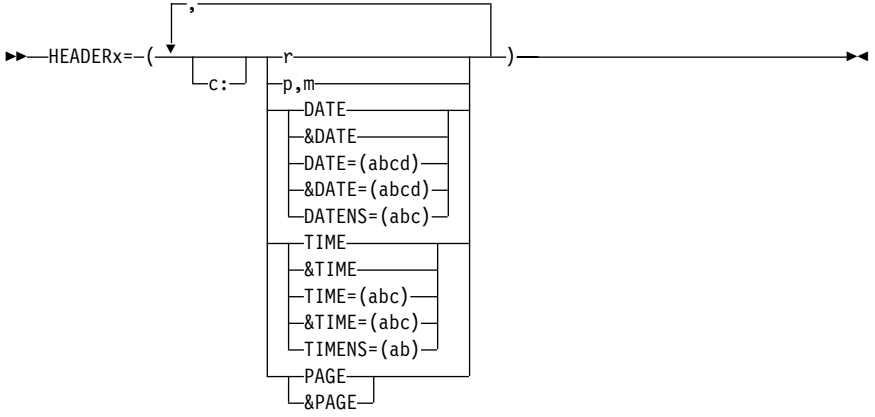
| Operand | Description |
|---|---|
| n | Lines per page. |

## HEADER1=

Specifies elements to appear in report header.

## HEADER2=

Specifies elements to appear in page header.



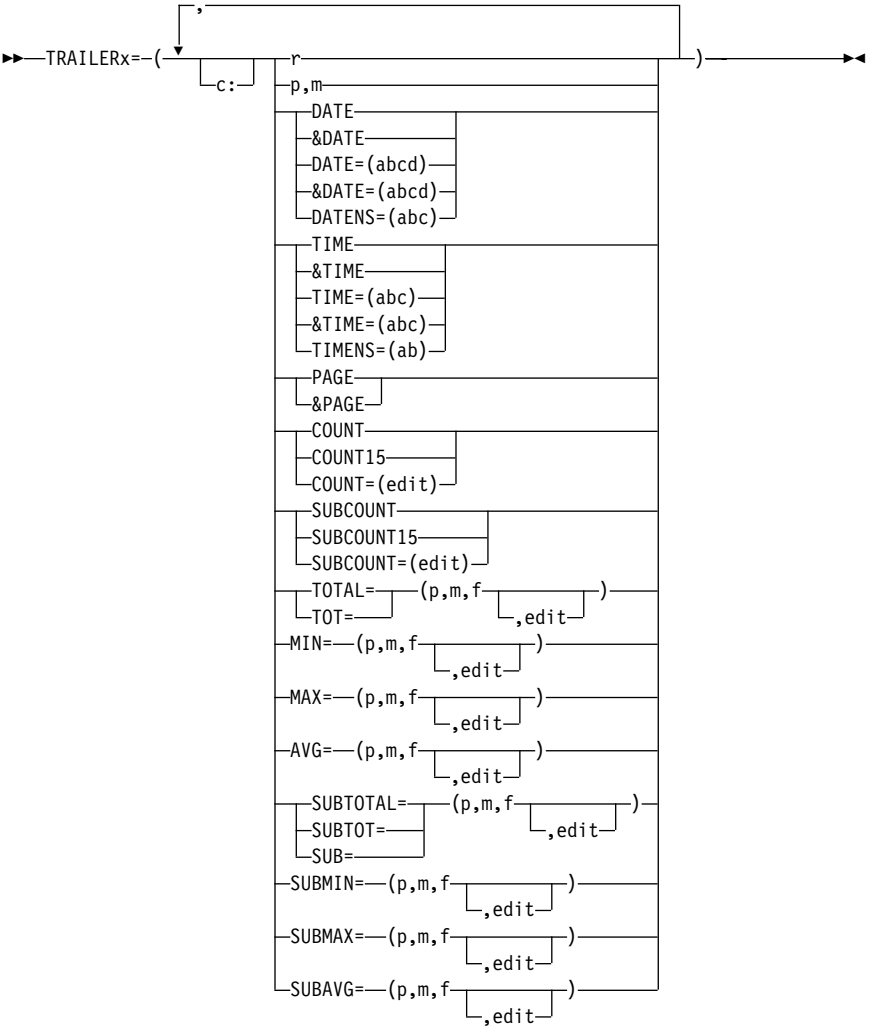

## TRAILER1=

Specifies elements to appear in report trailer.

## TRAILER2=

Specifies elements to appear in page trailer.

## OUTFIL—Creating Multiple Output Data Sets




| Operand | Description |
|---|---|
| c: | Column in which first position of report element appears. |
| r | Blanks or character string appear in report record, or new report record starts in header or trailer, with or without intervening blank lines. Permissible values: nX (n bytes of EBCDIC blanks); n'xx...x' (character string constant); /.../ or n/ (blank lines or new line). |

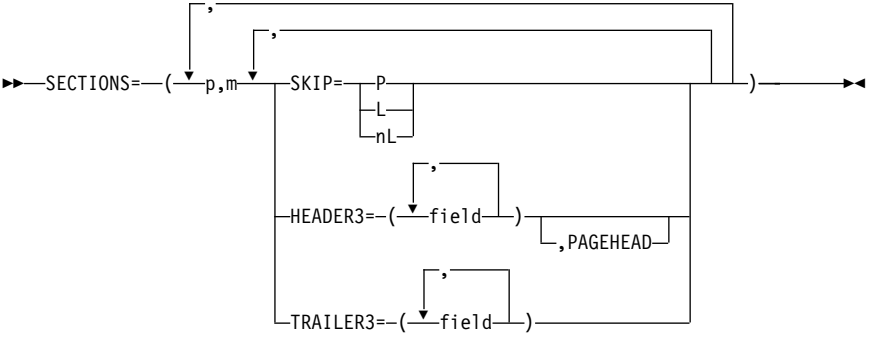
| Operand | Description |
|---|---|
| p,m | Unedited input field.<br>**p** Position of input field; a byte location within a record that is the first byte of an input field.<br>**m** Length of input field. |
| DATE | Current date in mm/dd/yy format. |
| &DATE | Can be used instead of DATE. |
| DATE=(abcd) | Current date in abcd format. |
| &DATE=(abcd) | Can be used instead of DATE=(abcd). |
| DATENS=(abc) | Current date in abc format. |
| TIME | Current time in hh:mm:ss format. |
| &TIME | Can be used instead of TIME. |
| TIME=(abc) | Current time in hhcmmcss xx format. |
| &TIME=(abc) | Can be used instead of TIME=abc. |
| TIMENS=(ab) | Current time in hhmmss xx format. |
| PAGE | Page number. |
| &PAGE | Can be used instead of PAGE. |
| COUNT | Count of input records as 8–digit value. |
| COUNT15 | Count of input records as 15–digit value. |
| COUNT= | Edited count of input records.<br>**edit** How 15–digit count is edited.<br>See p,m,f,edit under OUTREC= above. |
| SUBCOUNT | Running count of input records as 8–digit value. |
| SUBCOUNT15 | Running count of input records as 15–digit value. |
| SUBCOUNT= | Edited running count of input records.<br>**edit** How 15–digit running count is edited.<br>See p,m,f,edit under OUTREC= above. |
| TOTAL=<br>TOT= | Edited total for values of a numeric input field in all data records.<br>**p,m,f,edit**<br>Numeric input field for total, and how total is edited.<br>See p,m,f,edit under OUTREC= above. |

| Operand | Description |
|---|---|
| MIN= | Edited minimum for values of a numeric input field in all data records.<br>**p,m,f,edit**<br>Numeric input field for minimum, and how minimum is edited.<br>See p,m,f,edit under OUTREC= above. |
| MAX= | Edited maximum for values of a numeric input field in all data records.<br>**p,m,f,edit**<br>Numeric input field for maximum, and how maximum is edited.<br>See p,m,f,edit under OUTREC= above. |
| AVG= | Edited average for values of a numeric input field in all data records.<br>**p,m,f,edit**<br>Numeric input field for average, and how average is edited.<br>See p,m,f,edit under OUTREC= above. |
| SUBTOTAL=<br>SUBTOT=<br>SUB= | Edited running total for values of a numeric input field in all data records.<br>**p,m,f,edit**<br>Numeric input field for running total, and how running total is edited.<br>See p,m,f,edit under OUTREC= above. |
| SUBMIN= | Edited running minimum for values of a numeric input field in all data records.<br>**p,m,f,edit**<br>Numeric input field for running minimum, and how running minimum is edited.<br>See p,m,f,edit under OUTREC= above. |
| SUBMAX= | Edited running maximum for values of a numeric input field in all data records.<br>**p,m,f,edit**<br>Numeric input field for running maximum, and how running maximum is edited.<br>See p,m,f,edit under OUTREC= above. |

| Operand | Description |
|---|---|
| SUBAVG= | Edited running average for values of a numeric input field in all data records.<br>**p,m,f,edit**<br>Numeric input field for running average, and how running average is edited.<br>See p,m,f,edit under OUTREC= above. |

## SECTIONS=

Specifies section break processing.




| Operand | Description |
|---|---|
| p,m | Section break field in input records that divides report into sections.<br>**p** See p under HEADER1= above.<br>**m** See m under HEADER1= above. |
| SKIP= | Specifies either:<br>**P** Each section appears on new page.<br>**L** One blank line appears between sections on the same page.<br>**nL** n blank lines appear between sections on the same page. |
| HEADER3= | Specifies elements to appear in section header.<br>See descriptions under HEADER1= above. |
| PAGEHEAD | Specifies that section header appears at top of each report page, as well as before each section. |

## OUTFIL—Creating Multiple Output Data Sets

| Operand | Description |
|---|---|
| TRAILER3= | Specifies elements to appear in section trailer.<br><br>See descriptions under TRAILER1= above. |

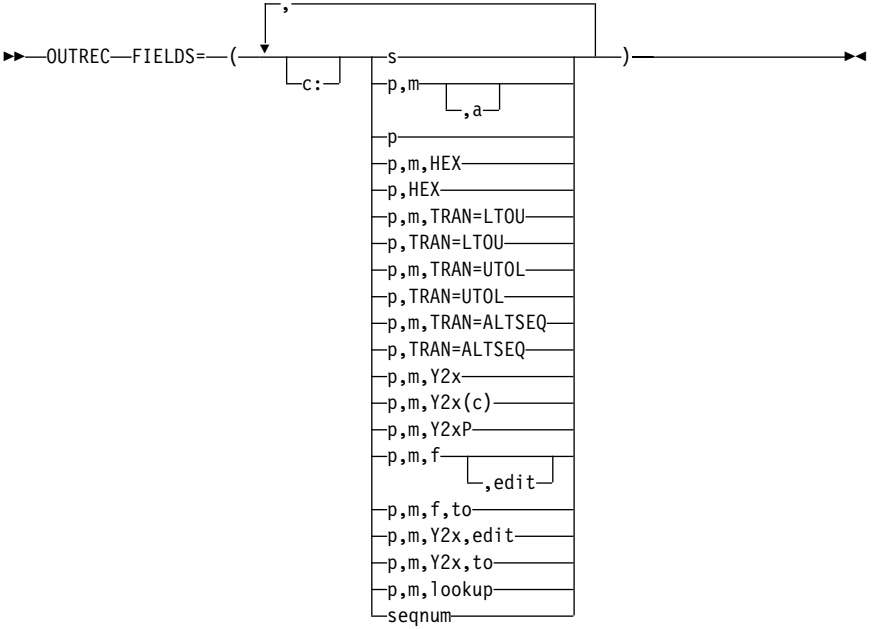
### NODETAIL

Shows headers and trailers with summarized data, but not the data itself.

### REMOVECC

Specifies that ANSI carriage control characters are to be removed from a report.

## OUTREC—Reformatting the Output Record



**Example**

```
OUTREC FIELDS=(20,4,2X,12,3,C'END')
```

Reformats output records, positions 20 through 23 and positions 12 through 14, into positions 1 through 4 and 7 through 9, respectively, inserts blanks in positions 5 and 6, and inserts the character string 'END' in positions 10 through 12.

## FIELDS=

Order of separation fields, unedited and edited input fields, and sequence numbers in the reformatted output record.

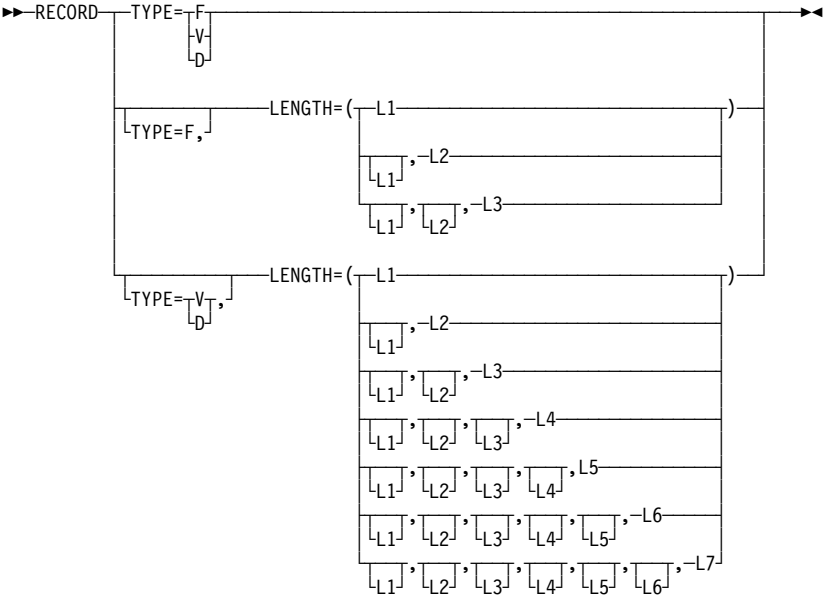
| Operand | Description |
|---|---|
| c: | Column for separation field, input field or sequence number in the reformatted output record. |
| s | A separation field appears in the reformatted output record. Permissible values: nX (blank separation); nZ (binary zero separation); nC'xx...x' (character string separation); nX'yy...yy' (hexadecimal string separation); DATEnP (packed decimal current date); DATEn and DATEn(c) (character current date); TIMEnP (packed decimal current time); TIMEn and TIMEn(c) (character current time). |
| p,m,a | Unedited input field.<br>**p** Position of input field; a byte location within a record that is the first byte of an input field.<br>**m** Length of input field.<br>**a** Alignment of input field in the reformatted output record. Permissible values: H (halfword aligned); F (fullword aligned); D (doubleword aligned). |
| p | Unedited variable part of input record.<br>see p under p,m,a above. |
| p,m,HEX | Hexadecimal representation of input field.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**HEX** Requests hexadecimal representation of input field. |

| Operand | Description |
|---|---|
| p,HEX | Hexadecimal representation of variable part of input record.<br>**p** See p under p,m,a above.<br>**HEX** Requests hexadecimal representation of variable part of input record. |
| p,m,TRAN=LTOU | Uppercase representation of input record.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**TRAN=LTOU**<br>Requests lowercase (a-z) to uppercase (A-Z) conversion. |
| p,TRAN=LTOU | Uppercase representation of variable part of input record.<br>**p** See p under p,m,a above.<br>**TRAN=LTOU**<br>Requests lowercase (a-z) to uppercase (A-Z)conversion. |
| p,m,TRAN=UTOL | Lowercase representation of input field.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**TRAN=UTOL**<br>Requests uppercase (A-Z) to lowercase (a-z). |
| p,TRAN=UTOL | Lowercase representation of variable part of input record.<br>**p** See p under p,m,a above.<br>**TRAN=UTOL**<br>Requests uppercase (A-Z) to lowercase (a-z) conversion. |
| p,m,TRAN=ALTSEQ | Representation of input field according to the ALTSEQ table.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**TRAN=LTOU**<br>Requests conversion of characters according to the ALTSEQ table. |

| Operand | Description |
|---|---|
| p,TRAN=ALTSEQ | Representation of variable part of input record according to the ALTSEQ table.<br>**p** See p under p,m,a above.<br>**TRAN=LTOU** Requests conversion of characters according to the ALTSEQ table. |
| p,m,Y2x | Four-digit year CH date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** Date format. |
| p,m,Y2x(c) | Four-digit year CH date representation with separators of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** Date format.<br>**c** Separator character. |
| p,m,Y2xP | Four-digit year PD date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2xP** Date format. |
| p,m,f,edit | Edited numeric input field.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**f** Format of numeric field: BI, FI, ZD, PD, PD0, FS, CSF, DTn, or TMn.<br>**edit** How field is edited.<br>See *Application Programming Guide* for details. |

| Operand | Description |
|---|---|
| p,m,f,to | Converted numeric input field.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**f** See f under p,m,f,edit above.<br>**to** How field is converted.<br>See *Application Programming Guide* for details. |
| p,m,Y2x,edit | Edited four-digit year CH date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** See Y2x under p,m,Y2x above.<br>**edit** See edit under p,m,f,edit above. |
| p,m,Y2x,to | Converted four-digit year date representation of two-digit year CH, ZD or PD input date field using the century window.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**Y2x** See Y2x under p,m,Y2x above.<br>**to** See to under p,m,f,to above. |
| p,m,lookup | Character or hexadecimal string from lookup table.<br>**p** See p under p,m,a above.<br>**m** See m under p,m,a above.<br>**lookup** How input field is changed using a lookup table.<br>See *Application Programming Guide* for details. |
| seqnum | Sequence number: BI, PD, ZD, FS or CSF.<br>See *Application Programming Guide* for details. |

## RECORD—Describing the Record Format and Length




**Example**

```
RECORD TYPE=V,LENGTH=(,100)
```

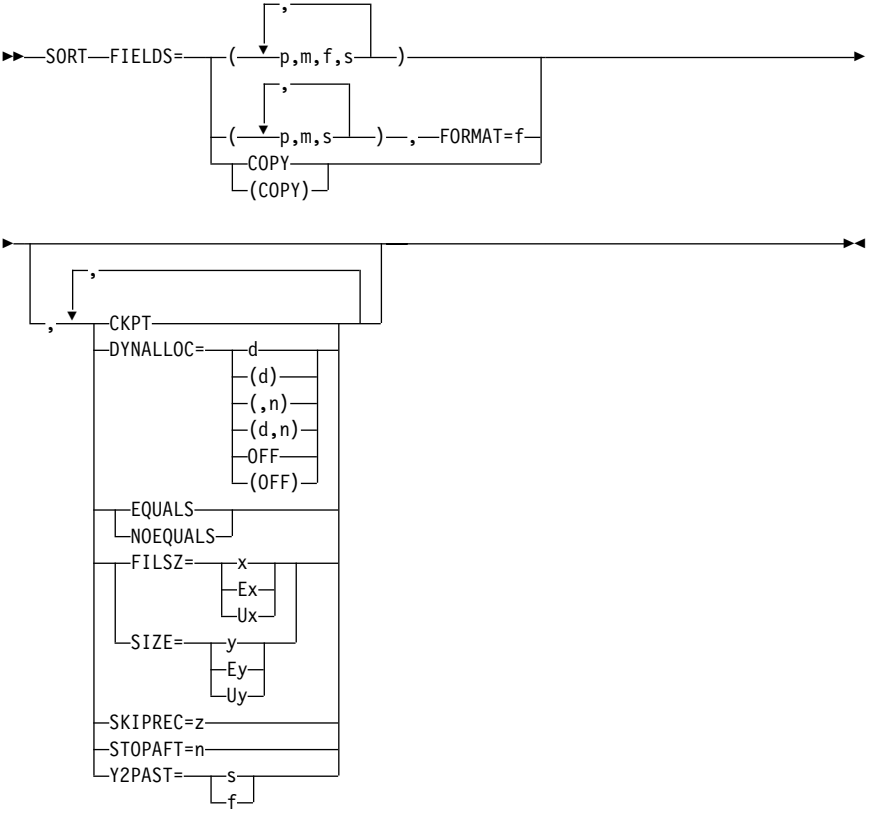
Variable-length record processing will be used. The E15 exit will pass back records with a maximum length of 100 bytes.

| Operand | Description |
|---|---|
| TYPE= | Record type. |
| | **F** Fixed-length records. |
| | **FB** Fixed-length records. Equivalent to F. |
| | **V** Variable-length EBCDIC records. |
| | **VB** Variable-length EBCDIC records. Equivalent to V. |
| | **D** Variable-length ISCII/ASCII records. |
| | **DB** Variable-length ISCII/ASCII records. Equivalent to D. |

## RECORD—Describing the Record Format and Length

| Operand | Description |
|---|---|
| LENGTH= | For fixed-length records:<br>**L1** Input record length.<br>**L2** Record length after E15.<br>**L3** Output record length.<br><br>For variable-length records:<br>**L1** Maximum input record length.<br>**L2** Maximum record length after E15.<br>**L3** Maximum output record length.<br>**L4** Minimum record length.<br>**L5** Average record length.<br>**L6** Accepted by DFSORT but not used; reserved for future use.<br>**L7** Accepted by DFSORT but not used; reserved for future use. |

## SORT—Specifying a SORT or COPY




SORT and MERGE control statements are mutually exclusive.

**Example**

```
SORT FIELDS=(2,5,CH,A)
```

The input record control field starts on byte 2, is 5 bytes long, contains character (EBCDIC) data, and is to be sorted in ascending order.

| Operand | Description |
|---|---|
| FIELDS= | Control fields must be described in descending order of significance.<br>**p** Position of the control field; a byte location within a record that is the first byte of a control field.<br>**m** Length of the control field.<br>**f** Format of the control field.<br>**s** Sequence of the data. Can be:<br>A — ascending<br>D — descending<br>E — user-modified control field (used with an E61 user exit) |
| FORMAT= | Can be used instead of the f subparameter of FIELDS= if all control field data formats are the same. |
| FIELDS=COPY<br>FIELDS=(COPY) | Copies a SORTIN data set to one or more output data sets. |
| CKPT | Activates the operating system's checkpoint/restart facility. |
| DYNALLOC | Allocates work data sets dynamically with values for d and n specified (or defaulted) by the DYNALOC option of the ICEMAC installation macro. |
| DYNALLOC= | Allocates work data sets dynamically.<br>**d** Specifies the device name.<br>**n** Specifies the maximum number of requested work data sets. |
| DYNALLOC=OFF | Prevents dynamic allocation of work data sets. |
| EQUALS | Specifies that the original sequence of records that collate identically must be preserved. |
| NOEQUALS | Specifies that the original sequence of records that collate identically need not be preserved. |
| FILSZ= | Specifies the number of records to be processed.<br>**x** Is the exact number of records to be processed.<br>**Ex** Is the estimated number of records to be processed. x must be immediately preceded by the letter E.<br>**Ux** Is the number of records to be processed. x must be immediately preceded by the letter U. |

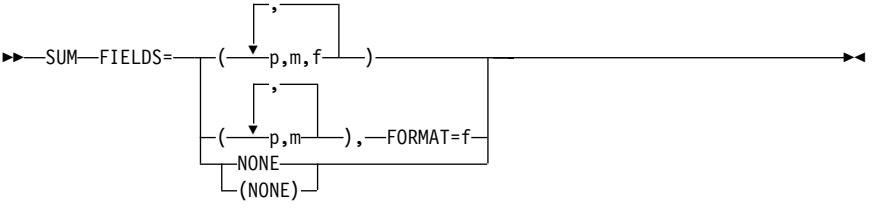
| Operand | Description |
|---|---|
| SIZE= | Specifies the number of records in the input data sets.<br>**y** Is the exact number of records in the input data sets.<br>**Ey** Is the estimated number of records in the input data sets. y must be immediately preceded by the letter E.<br>**Uy** Is the number of records in the input data sets. y must be immediately preceded by the letter U. |
| SKIPREC= | Skips z input records before DFSORT begins to sort or copy. |
| STOPAFT= | Indicates the number of records you want accepted for sorting or copying. |
| Y2PAST= | Specifies the sliding or fixed century window.<br>**s** Starts the sliding century window at the current year − s.<br>**f** Starts the fixed century window at f.<br>**Note:** CENTURY and CENTWIN can be used instead of Y2PAST. |

## Control Field Formats for SORT

| Format | Length | Description |
|---|---|---|
| CH | 1 to 4092 bytes | Unsigned character (if CHALT is in effect, CH is treated as AQ) |
| AQ | 1 to 4092 bytes | Character, alternate collating sequence |
| ZD | 1 to 32 bytes | Signed zoned decimal |
| PD | 1 to 32 bytes | Signed packed decimal |
| PD0 | 2 to 8 bytes | Packed decimal with sign and first digit ignored |
| FI | 1 to 256 bytes | Signed fixed point |
| BI | 1 bit to 4092 bytes | Unsigned binary |
| FL | 1 to 256 bytes | Signed floating point |
| AC | 1 to 4092 bytes | ISCII/ASCII character |
| CSF or FS | 1 to 16 bytes | Signed numeric, optional leading floating sign. |

| Format | Length | Description |
|---|---|---|
| CSL or LS | 2 to 256 bytes | Signed numeric, leading separate sign |
| CST or TS | 2 to 256 bytes | Signed numeric, trailing separate sign |
| CLO or OL | 1 to 256 bytes | Signed numeric, leading overpunch sign |
| CTO or OT | 1 to 256 bytes | Signed numeric, trailing overpunch sign |
| ASL | 2 to 256 bytes | Signed ISCII/ASCII numeric, leading separate sign |
| AST | 2 to 256 bytes | Signed ISCII/ASCII numeric, trailing separate sign |
| D1 | 1 to 4092 bytes | User-defined data type (requires an EFS program) |
| Y2x | 1 to 6 bytes | Date format |

## SUM—Adding Summary Fields




**Example**

```
SUM FIELDS=(41,8,ZD,49,4,FI)
```

This statement designates an 8-byte zoned decimal field at byte 41 and a 4-byte fixed integer field at byte 49 as summary fields.

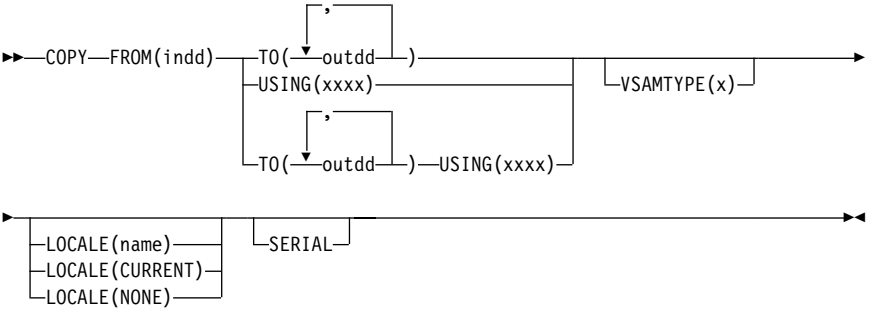
| Operand | Description |
|---|---|
| FIELDS= | Specifies input record numeric fields as summary fields.<br>**p** Position of summary field; a byte location within a record that is the first byte of a summary field.<br>**m** Length of summary field.<br>**f** Data format of summary field. |
| FIELDS=NONE<br>FIELDS=(NONE) | Eliminates records with duplicate keys; no summing is done. |
| FORMAT= | Can be used instead of the f subparameter of FIELDS= if all summary field data formats are the same. |

## Summary Field Formats for SUM

| Format | Length | Description |
|---|---|---|
| BI | 2, 4, or 8 bytes | Unsigned binary |
| FI | 2, 4, or 8 bytes | Signed fixed point |
| FL | 4, 8, or 16 bytes | Signed floating point |
| PD | 1 to 16 bytes | Signed packed decimal |
| ZD | 1 to 18 bytes | Signed zoned decimal |

# Chapter 3. ICETOOL Operators

## COPY—Copying Data Sets




**Example**

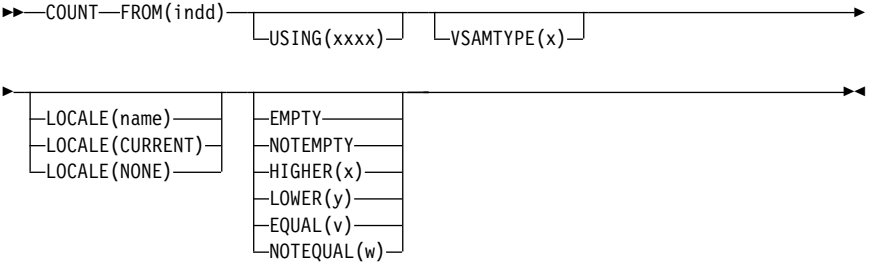
```
COPY FROM(IN1) TO (OUT1,OUT2,OUT3) USING(CTL1)
```

Copies the records from the IN1 data set to the OUT1, OUT2, and OUT3 data sets using the DFSORT control statements in the CTL1CNTL data set.

| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| TO(outdd,...) | Specifies the ddnames of the output data sets. |
| USING(xxxx) | Specifies the first 4 characters of the xxxxCNTL ddname for the DFSORT control statement data set. |
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br><br>Valid values for x are:<br><br>**F** Fixed-length records.<br><br>**V** Variable-length records. |
| LOCALE(name) | Specifies locale processing and designates name of active locale. |
| LOCALE(CURRENT) | Specifies locale processing and keeps current locale active. |

| Operand | Description |
|---|---|
| LOCALE(NONE) | Specifies locale processing is not used. |
| SERIAL | Specifies OUTFIL processing is not used when multiple outdd data sets are specified. |

## COUNT—Counting Records




**Example**

```
COUNT FROM(IN2)
```

Prints a message giving the count of records in the IN2 data set.

| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| USING(xxxx) | Specifies the first 4 characters of the xxxxCNTL ddname for the DFSORT control statement data set. |
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br><br>Valid values for x are:<br><br>**F** Fixed-length records.<br><br>**V** Variable-length records. |
| LOCALE(name) | Specifies locale processing and designates name of active locale. |
| LOCALE(CURRENT) | Specifies locale processing and keeps current locale active. |
| LOCALE(NONE) | Specifies locale processing is not used. |
| EMPTY | Sets RC=12 if the input data set is empty. |

| Operand | Description |
|---|---|
| NOTEMPTY | Sets RC=12 if the input data set is not empty. |
| HIGHER(x) | Sets RC=12 if the record count is higher than x. |
| LOWER(y) | Sets RC=12 if the record count is lower than y. |
| EQUAL(v) | Sets RC=12 if the record count is equal to v. |
| NOTEQUAL(w) | Sets RC=12 if the record count is not equal to w. |

## DEFAULTS—Displaying Installation Defaults



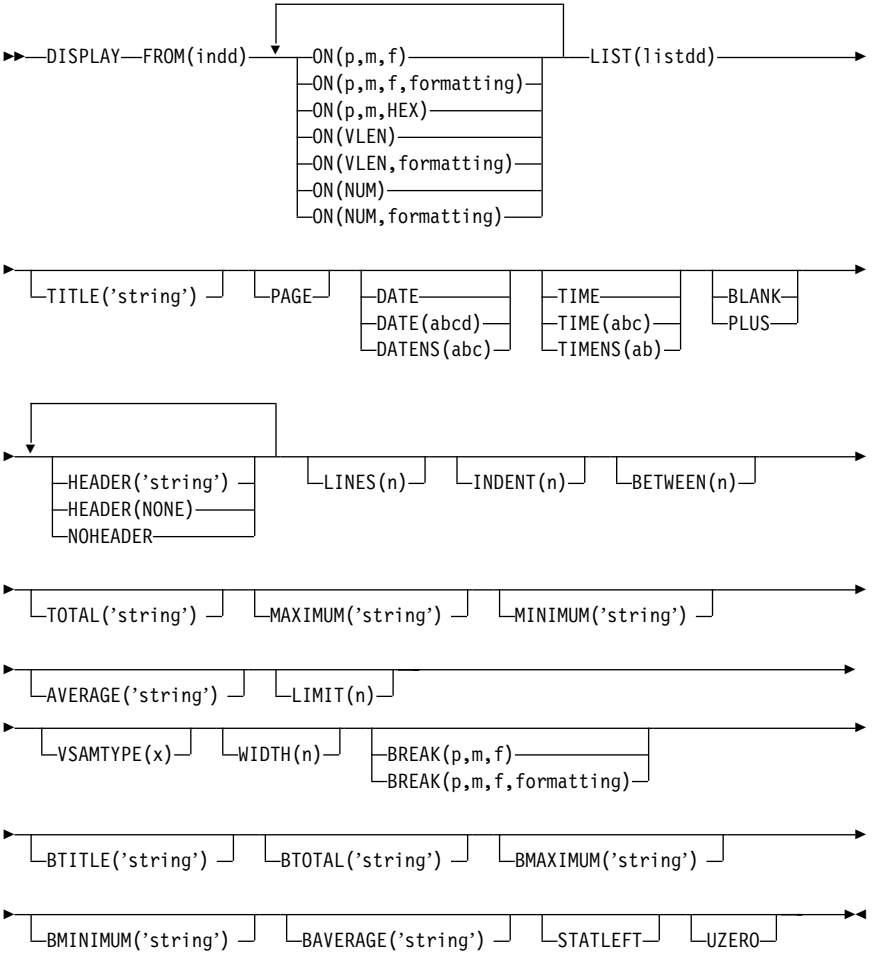

**Example**

```
DEFAULTS LIST(OPTIONS)
```

Prints, in the OPTIONS data set, the DFSORT installation defaults.

| Operand | Description |
|---|---|
| LIST(listdd) | Specifies the ddname of the list data set. |

## DISPLAY—Printing Reports



**Example**

```
DISPLAY FROM(IN) LIST(LIST1) -
      TITLE('National Accounting Report') -
      PAGE DATE TIME -
      HEADER('Division') HEADER('Revenue') HEADER('Profit/Loss') -
      ON(1,25,CH)        ON(45,10,ZD)       ON(35,10,ZD) -
      BLANK -
      TOTAL('Company Totals') -
      AVERAGE('Company Averages')
```

Prints, in the LIST1 data set:

- A title line containing the specified title, the page number, the date and the time
- A heading line containing the specified underlined headings
- Data lines in the BLANK format containing:
  - The characters from positions 1-25 of the IN data set
  - The zoned decimal values from positions 45-54 of the IN data set
  - The zoned decimal values from positions 35-44 of the IN data set
- A TOTAL line containing the specified string and the total for each of the two zoned decimal fields in the BLANK format
- An AVERAGE line containing the specified string and the average for each of the two zoned decimal fields in the BLANK format.

| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| ON(p,m,f) | Specifies the position, length, and format of a numeric or character field.<br>**p** Position of the field; the first byte of the field relative to the beginning of the input record.<br>**m** Length of the field.<br>**f** Format of the field. |

| Operand | Description |
|---|---|
| ON(p,m,f,formatting) | Specifies the position, length, and format of a numeric or character field and how the data is formatted.<br>**p,m,f** See ON(p,m,f) above.<br>**formatting**<br>Items that indicate how data is formatted for printing.<br>See *Application Programming Guide* for details. |
| ON(p,m,HEX) | Specifies the position and length of a character field to be printed in hexadecimal format (00–FF for each byte).<br>**p** Position of the field; the first byte of the field relative to the beginning of the input record.<br>**m** Length of the field. |
| ON(VLEN) | Same as ON(1,2,BI); specifies a two-byte binary field starting at position 1. For variable-length records, ON(VLEN) represents the record length for each input record. |
| ON(VLEN,formatting) | Same as ON(1,2,BI,formatting); specifies a two-byte binary field starting at position 1, and how the data is formatted. For variable-length records, ON(VLEN,formatting) represents the record length for each input record.<br>**formatting**<br>Items that indicate how data is formatted for printing.<br>See *Application Programming Guide* for details. |
| ON(NUM) | Record number. Starts at 1 and is incremented by 1 for each input record. |
| ON(NUM,formatting) | Record number, and how it is formatted. Starts at 1 and is incremented by 1 for each input record.<br>**formatting**<br>Items that indicate how data is formatted for printing.<br>See *Application Programming Guide* for details. |
| LIST(listdd) | Specifies the ddname of the list data set. |
| TITLE('string') | Specifies printing of a title string in the title line. |

| Operand | Description |
|---|---|
| PAGE | Specifies printing of the page number in the title line. |
| DATE | Specifies printing of the date in the title line in the form mm/dd/yy. |
| DATE(abcd) | Specifies printing of the date in the title line in the form adbdc. |
| DATENS(abc) | Specifies printing of the date in the title line in the form abc. |
| TIME | Specifies printing of the time in the title line in the form hh:mm:ss. |
| TIME(abc) | Specifies printing of the time in the title line in the form hhcmmcss xx. |
| TIMENS(ab) | Specifies printing of the time in the title line in the form hhmmss xx. |
| BLANK | Specifies an alternate format for printing character and numeric data using blank for plus sign, − for minus sign, and no leading zeros. Column widths are dynamically adjusted and headings and numeric values are right justified. |
| PLUS | Specifies an alternate format for printing character and numeric data using + for plus sign, − for minus sign, and no leading zeros. Column widths are dynamically adjusted and headings and numeric values are right justified. |
| HEADER('string') | Specifies that a heading is to be printed for the corresponding ON field. |
| HEADER(NONE) | Specifies that a heading is not to be printed for the corresponding ON field. |
| NOHEADER | Specifies that headings for ON fields are not to be printed. |
| LINES(n) | Specifies the number of lines per page for the list data set. |
| INDENT(n) | Specifies the number of blanks for indentation. |
| BETWEEN(n) | Specifies the number of blanks between columns. |
| TOTAL('string') | Specifies that an overall TOTAL line is to be printed after the columns of data. |
| MAXIMUM('string') | Specifies that an overall MAXIMUM line is to be printed after the columns of data. |
| MINIMUM('string') | Specifies that an overall MINIMUM line is to be printed after the columns of data. |
| AVERAGE('string') | Specifies that an overall AVERAGE line is to be printed after the columns of data. |

| Operand | Description |
|---|---|
| LIMIT(n) | Specifies the maximum number of invalid decimal values (the default is 200). If n invalid values are found, ICETOOL terminates this operation. |
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br><br>Valid values for x are:<br><br>**F** Fixed-length records.<br><br>**V** Variable-length records. |
| WIDTH(n) | Specifies the line length and LRECL for the list data set. |
| BREAK(p,m,f) | Specifies a numeric or character break field dividing the report into sections.<br><br>**p,m,f** See ON(p,m,f) above. |
| BREAK(p,m,f,formatting) | Specifies a numeric or character break field dividing the report into sections, and how the data is formatted..<br><br>**p,m,f** See BREAK(p,m,f) above.<br><br>**formatting**<br>Items that indicate how data is formatted for printing.<br><br>See *Application Programming Guide* for details. |
| BTITLE('string') | Specifies a string in the break title line for each section. |
| BTOTAL('string') | Specifies a break TOTAL line for each section. |
| BMAXIMUM('string') | Specifies a break MAXIMUM line for each section. |
| BMINIMUM('string') | Specifies a break MINIMUM line for each section. |
| BAVERAGE('string') | Specifies a break AVERAGE line for each section. |
| STATLEFT | Specifies that strings for statistics are placed to the left of the first data column. |
| UZERO | Treats − 0 and + 0 as signed, that is, as the same. |

## Field Formats for DISPLAY

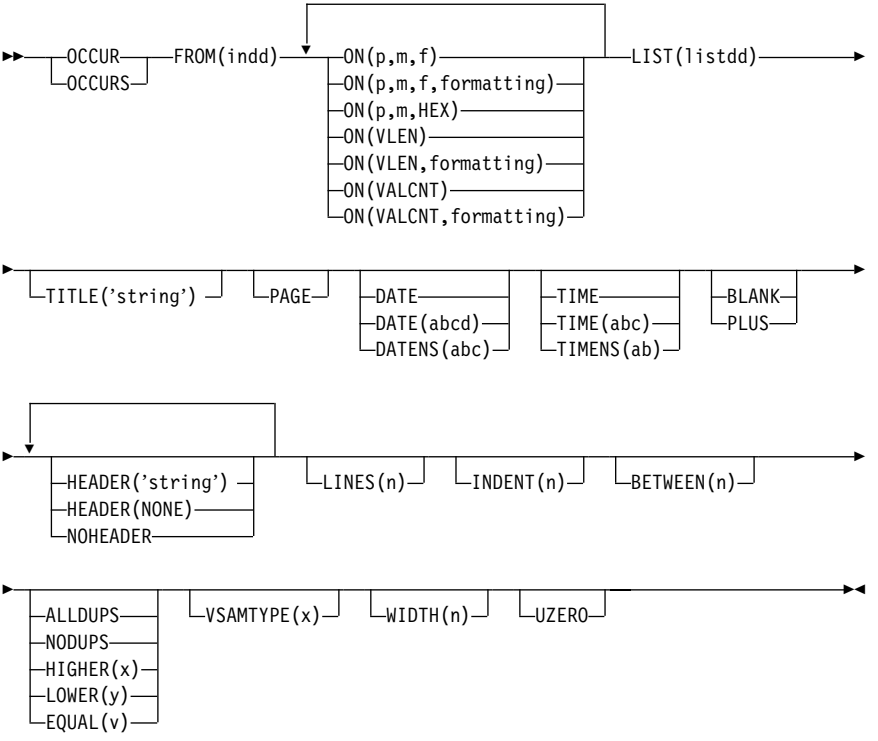
| Format | Length | Description |
|---|---|---|
| BI | 1 to 4 bytes | Unsigned binary |
| FI | 1 to 4 bytes | Signed fixed-point |
| PD | 1 to 8 bytes | Signed packed decimal |

| Format | Length | Description |
|---|---|---|
| ZD | 1 to 15 bytes | Signed zoned decimal |
| CH | 1 to 1500 bytes | Character |
| CSF or FS | 1 to 16 bytes (15 digit limit) | Signed numeric with optional leading floating sign |
| DT1 | 4 bytes | SMF date as Z'yyyymmdd' |
| DT2 | 4 bytes | SMF date as Z'yyyymm' |
| DT3 | 4 bytes | SMF date as Z'yyyyddd' |
| TM1 | 4 bytes | SMF time as Z'hhmmss' |
| TM2 | 4 bytes | SMF time as Z'hhmm' |
| TM3 | 4 bytes | SMF time as Z'hh' |
| TM4 | 4 bytes | SMF time as Z'hhmmssxx' |

## MODE—Setting the Mode


| Operand | Description |
|---|---|
| STOP | Stops subsequent operations and sets SCAN mode in effect if an operation fails. |
| CONTINUE | Continues with subsequent operations even if an operation fails. |
| SCAN | Checks ICETOOL statements for errors. |

## OCCUR—Reporting Counts of Unique Values




**Example**

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

Prints, in the VOLSERS data set:

- A heading line containing the standard headings
- A data line for each unique ON(40,6,CH) value in the standard format containing:
  - The characters from positions 40-45 of the SOURCE data set for the unique value
  - The count of occurrences in the SOURCE data set of the unique value

| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| ON(p,m,f) | Specifies the position, length, and format of a numeric or character field.<br>**p** Position of the field; the first byte of the field relative to the beginning of the input record.<br>**m** Length of the field.<br>**f** Format of the field. |
| ON(p,m,f,formatting) | Specifies the position, length, and format of a numeric or character field and how the data is formatted.<br>**p,m,f** See ON(p,m,f) above.<br>**formatting** Items that indicate how data is formatted for printing.<br>See *Application Programming Guide* for details. |
| ON(p,m,HEX) | Specifies the position and length of a character field to be printed in hexadecimal format (00–FF for each byte).<br>**p** Position of the field; the first byte of the field relative to the beginning of the input record.<br>**m** Length of the field. |
| ON(VLEN) | Same as ON(1,2,BI); specifies a two-byte binary field starting at position 1. For variable-length records, ON(VLEN) represents the record length for each input record. |
| ON(VLEN,formatting) | Same as ON(1,2,BI,formatting); specifies a two-byte binary field starting at position 1, and how the data is formatted. For variable-length records, ON(VLEN,formatting) represents the record length for each input record.<br>**formatting** Items that indicate how data is formatted for printing.<br>See *Application Programming Guide* for details. |
| ON(VALCNT) | Specifies that the number of occurrences of each unique value is to be printed. |

## OCCUR—Reporting Counts of Unique Values

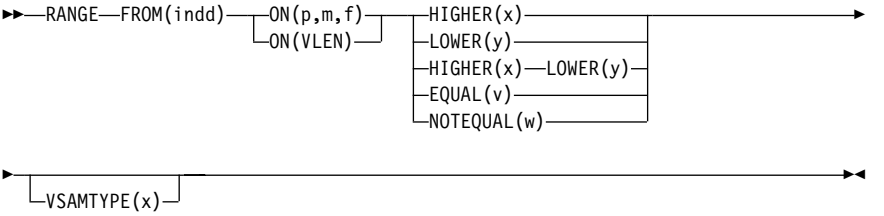
| Operand | Description |
|---|---|
| ON(VALCNT,formatting) | Specifies that the number of occurrences of each unique value is to be printed, and how the data is formatted.<br>**formatting**<br>Items that indicate how data is formatted for printing.<br>See *Application Programming Guide* for details. |
| LIST(listdd) | Specifies the ddname of the list data set. |
| TITLE('string') | Specifies printing of a title string in the title line. |
| PAGE | Specifies printing of the page number in the title line. |
| DATE | Specifies printing of the date in the title line in the form mm/dd/yy. |
| DATE(abcd) | Specifies printing of the date in the title line in the form adbdc. |
| DATENS(abc) | Specifies printing of the date in the title line in the form abc. |
| TIME | Specifies printing of the time in the title line in the form hh:mm:ss. |
| TIME(abc) | Specifies printing of the time in the title line in the form hhcmmcss xx. |
| TIMENS(ab) | Specifies printing of the time in the title line in the form hhmmss xx. |
| BLANK | Specifies an alternate format for printing character and numeric data using blank for plus sign, − for minus sign, and no leading zeros. Column widths are dynamically adjusted and headings and numeric values are right justified. |
| PLUS | Specifies an alternate format for printing character and numeric data using + for plus sign, − for minus sign, and no leading zeros. Column widths are dynamically adjusted and headings and numeric values are right justified. |
| HEADER('string') | Specifies that a heading is to be printed for the corresponding ON field. |
| HEADER(NONE) | Specifies that a heading is not to be printed for the corresponding ON field. |
| NOHEADER | Specifies that headings for ON fields are not to be printed. |
| LINES(n) | Specifies the number of lines per page for the list data set. |

| Operand | Description |
|---|---|
| INDENT(n) | Specifies the number of blanks for indentation. |
| BETWEEN(n) | Specifies the number of blanks between columns. |
| ALLDUPS | Limits the ON values printed to those that occur more than once. |
| NODUPS | Limits the ON values printed to those that occur only once. |
| HIGHER(x) | Limits the ON values that occur more than x times. |
| LOWER(y) | Limits the ON values printed to those that occur less than y times. |
| EQUAL(v) | Limits the ON values printed to those that occur v times. |
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br><br>Valid values for x are:<br><br>**F** Fixed-length records.<br><br>**V** Variable-length records. |
| WIDTH(n) | Specifies the line length and LRECL for the list data set. |
| UZERO | Treats − 0 and + 0 as signed, that is, as the same. |

## Field Formats for OCCUR

| Format | Length | Description |
|---|---|---|
| BI | 1 to 4 bytes | Unsigned binary |
| FI | 1 to 4 bytes | Signed fixed-point |
| PD | 1 to 8 bytes | Signed packed decimal |
| ZD | 1 to 15 bytes | Signed zoned decimal |
| CH | 1 to 1500 bytes | Character |
| CSF or FS | 1 to 16 bytes (15 digit limit) | Signed numeric with optional leading floating sign |
| DT1 | 4 bytes | SMF date as Z'yyyymmdd' |
| DT2 | 4 bytes | SMF date as Z'yyyymm' |
| DT3 | 4 bytes | SMF date as Z'yyyyddd' |
| TM1 | 4 bytes | SMF time as Z'hhmmss' |
| TM2 | 4 bytes | SMF time as Z'hhmm' |
| TM3 | 4 bytes | SMF time as Z'hh' |
| TM4 | 4 bytes | SMF time as Z'hhmmssxx' |

## RANGE—Counting Values in a Range




**Example**

```
RANGE FROM(DATA3) ON(29001,4,FI) -
HIGHER(-10000) LOWER(27)
```

Prints a message giving the count of fixed-point values from positions 29001-29004 of the DATA3 data set that are higher than -10 000 but lower than 27 (-10 000<value<27).

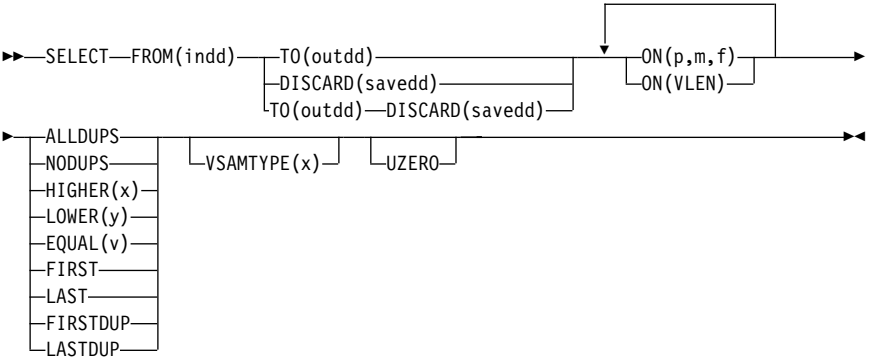
| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| ON(p,m,f) | Specifies the position, length, and format of the numeric field to be used for this operation.<br>**p** Position of the field; the first byte of the field relative to the beginning of the input record.<br>**m** Length of the field.<br>**f** Format of the field. |
| ON(VLEN) | Same as ON(1,2,BI); specifies a two-byte binary field starting at position 1. For variable-length records, ON(VLEN) represents the record length for each input record. |
| HIGHER(x) | Counts only those values higher than x as being in the range. |
| LOWER(y) | Counts only those values lower than y as being in the range. |
| EQUAL(v) | Counts only those values equal to v as being in the range. |

| Operand | Description |
|---|---|
| NOTEQUAL(w) | Counts only those values not equal to w as being in the range. |
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br><br>Valid values for x are:<br><br>**F** Fixed-length records.<br><br>**V** Variable-length records. |

## Field Formats for RANGE

| Format | Length | Description |
|---|---|---|
| BI | 1 to 4 bytes | Unsigned binary |
| FI | 1 to 4 bytes | Signed fixed-point |
| PD | 1 to 8 bytes | Signed packed decimal |
| ZD | 1 to 15 bytes | Signed zoned decimal |
| CSF or FS | 1 to 16 bytes (15 digit limit) | Signed numeric with optional leading floating sign |

## SELECT—Selecting Records by Occurrences of Fields




**Example**

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

Sorts the INPUT data set to the DUPS data set, selecting only the records from INPUT with characters in positions 11-18 and characters in positions 30-73 that occur more than once (that is, only records with duplicate ON field values).

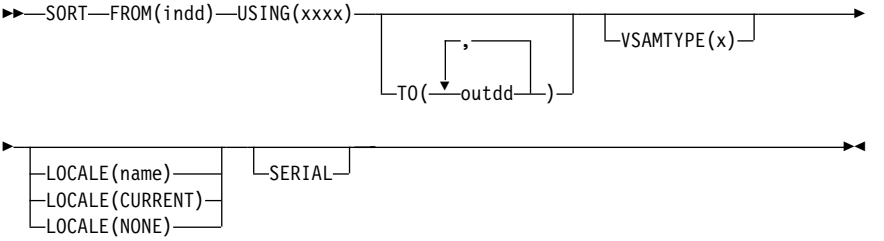
| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| TO(outdd) | Specifies the ddname of the output data set for the records that are selected. |
| DISCARD(savedd) | Specifies the ddname of the output data set for the records that are not selected. |
| ON(p,m,f) | Specifies the position, length, and format of a numeric or character field.<br>**p** Position of the field; the first byte of the field relative to the beginning of the input record.<br>**m** Length of the field.<br>**f** Format of the field. |
| ON(VLEN) | Same as ON(1,2,BI); specifies a two-byte binary field starting at position 1. For variable-length records, ON(VLEN) represents the record length for each input record. |

| Operand | Description |
|---|---|
| ALLDUPS | Limits the records selected to those with ON values that occur more than once. |
| NODUPS | Limits the records selected to those with ON values that occur only once. |
| HIGHER(x) | Limits the records selected to those with ON values that occur more than x times. |
| LOWER(y) | Limits the records selected to those with ON values that occur less than y times. |
| EQUAL(v) | Limits the records selected to those with ON values that occur v times. |
| FIRST | Limits the records selected to those with ON values that occur only once and the first record of those with ON values that occur more than once. |
| LAST | Limits the records selected to those with ON values that occur only once and the last record of those with ON values that occur more than once. |
| FIRSTDUP | Limits the records selected to the first record of those with ON values that occur more than once. |
| LASTDUP | Limits the records selected to the last record of those with ON values that occur more than once. |
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br><br>Valid values for x are:<br><br>**F** Fixed-length records.<br><br>**V** Variable-length records. |
| UZERO | Treats − 0 and + 0 as signed, that is, as the same. |

## Field Formats for SELECT

| Format | Length | Description |
|---|---|---|
| BI | 1 to 1500 bytes | Unsigned binary |
| FI | 1 to 256 bytes | Signed fixed-point |
| PD | 1 to 8 bytes | Signed packed decimal |
| ZD | 1 to 15 bytes | Signed zoned decimal |
| CH | 1 to 1500 bytes | Character |
| CSF or FS | 1 to 16 bytes | Signed numeric with optional leading floating sign |

## SORT—Sorting Data Sets



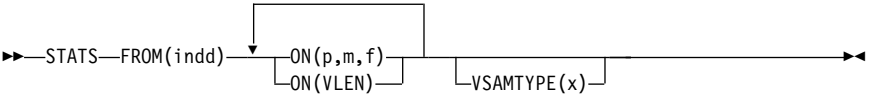

**Example**

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

Sorts the records from the IN1 data set to the OUT1, OUT2, and OUT3 data sets using the DFSORT control statements in the ABCDCNTL data set.

| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| USING(xxxx) | Specifies the first 4 characters of the xxxxCNTL ddname for the DFSORT control statement data set. |
| TO(outdd,...) | Specifies the ddnames of the output data sets. |
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br><br>Valid values for x are:<br><br>**F** Fixed-length records.<br><br>**V** Variable-length records. |
| LOCALE(name) | Specifies locale processing and designates name of active locale. |
| LOCALE(CURRENT) | Specifies locale processing and keeps current locale active. |
| LOCALE(NONE) | Specifies locale processing is not used. |
| SERIAL | Specifies OUTFIL processing is not used when multiple outdd data sets are specified. |

## STATS—Computing Statistics




**Example**

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

Prints messages giving the minimum, maximum, average, and total of the binary values in positions 1-2 of the DATA1 data set. For variable-length records, this gives statistics about the length of the records.

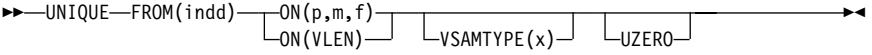
Also prints messages containing the minimum, maximum, average, and total of the zoned decimal values in positions 15-18 of the DATA1 data set.

| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| ON(p,m,f) | Specifies the position, length, and format of a numeric field to be used for this operation.<br>**p** Position of the field; the first byte of the field relative to the beginning of the input record.<br>**m** Length of the field.<br>**f** Format of the field. |
| ON(VLEN) | Same as ON(1,2,BI); specifies a two-byte binary field starting at position 1. For variable-length records, ON(VLEN) represents the record length for each input record. |
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br>Valid values for x are:<br>**F** Fixed-length records.<br>**V** Variable-length records. |

## Field Formats for STATS

| Format Code | Length | Description |
|---|---|---|
| BI | 1 to 4 bytes | Unsigned binary |
| FI | 1 to 4 bytes | Signed fixed-point |
| PD | 1 to 8 bytes | Signed packed decimal |
| ZD | 1 to 15 bytes | Signed zoned decimal |
| CSF or FS | 1 to 16 bytes | Signed numeric with optional leading floating sign |

## UNIQUE—Counting Unique Values




**Example**

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

Prints a message giving the count of unique zoned decimal values in positions 5-7 of the DATAIN data set.

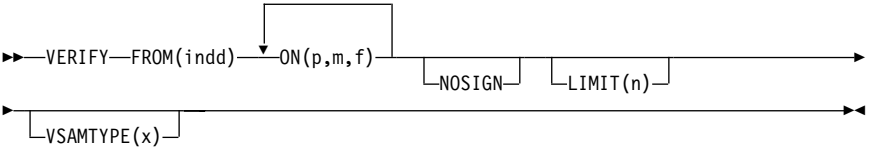
| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| ON(p,m,f) | Specifies the position, length, and format of a numeric or character field to be used for this operation.<br>**p** Position of the field; the first byte of the field relative to the beginning of the input record.<br>**m** Length of the field.<br>**f** Format of the field. |
| ON(VLEN) | Same as ON(1,2,BI); specifies a two-byte binary field starting at position 1. For variable-length records, ON(VLEN) represents the record length for each input record. |

| Operand | Description |
|---|---|
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br><br>Valid values for x are:<br><br>**F** Fixed-length records.<br><br>**V** Variable-length records. |
| UZERO | Treats − 0 and + 0 as signed, that is, as the same. |

## Field Formats for UNIQUE

| Format Code | Length | Description |
|---|---|---|
| BI | 1 to 256 bytes | Unsigned binary |
| FI | 1 to 256 bytes | Signed fixed-point |
| PD | 1 to 32 bytes | Signed packed decimal |
| ZD | 1 to 32 bytes | Signed zoned decimal |
| CH | 1 to 1500 bytes | Character |
| CSF or FS | 1 to 16 bytes | Signed numeric with optional leading floating sign |

## VERIFY—Verifying Decimal Values




**Example**

```
VERIFY FROM(NEW1) ON(22,16,PD) ON(7,9,ZD) LIMIT(10)
```

Checks for invalid digits (A-F) and invalid signs (0-9) in the packed decimal values from positions 22-37 and the zoned decimal values from positions 7-15 of the NEW1 data set. Terminates operation if 10 invalid values are found.

## VERIFY—Verifying Decimal Values

| Operand | Description |
|---|---|
| FROM(indd) | Specifies the ddname of the input data set. |
| ON(p,m,f) | Specifies the position, length, and format of a decimal field.<br>**p** Position of the field; the first byte of the field relative to the beginning of the input record.<br>**m** Length of the field.<br>**f** Format of the field. |
| NOSIGN | Specifies that the sign of the decimal values is not checked for validity. |
| LIMIT(n) | Specifies the maximum number of valid decimal values (the default is 200). If n invalid values are found ICETOOL terminates this operation. |
| VSAMTYPE(x) | Specifies the record format for a VSAM input data set.<br>Valid values for x are:<br>**F** Fixed-length records.<br>**V** Variable-length records. |

## Field Formats for VERIFY

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

# Chapter 4. DFSORT Job Control Statements

Details on coding DFSORT job control language (JCL) statements are explained in *Application Programming Guide*

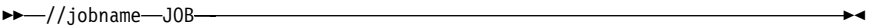



*Figure 2. //jobname Job Control Statement*

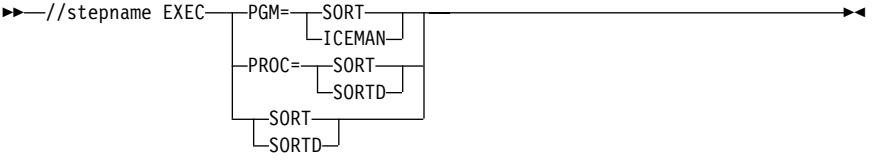



*Figure 3. //stepname Job Control Statement*

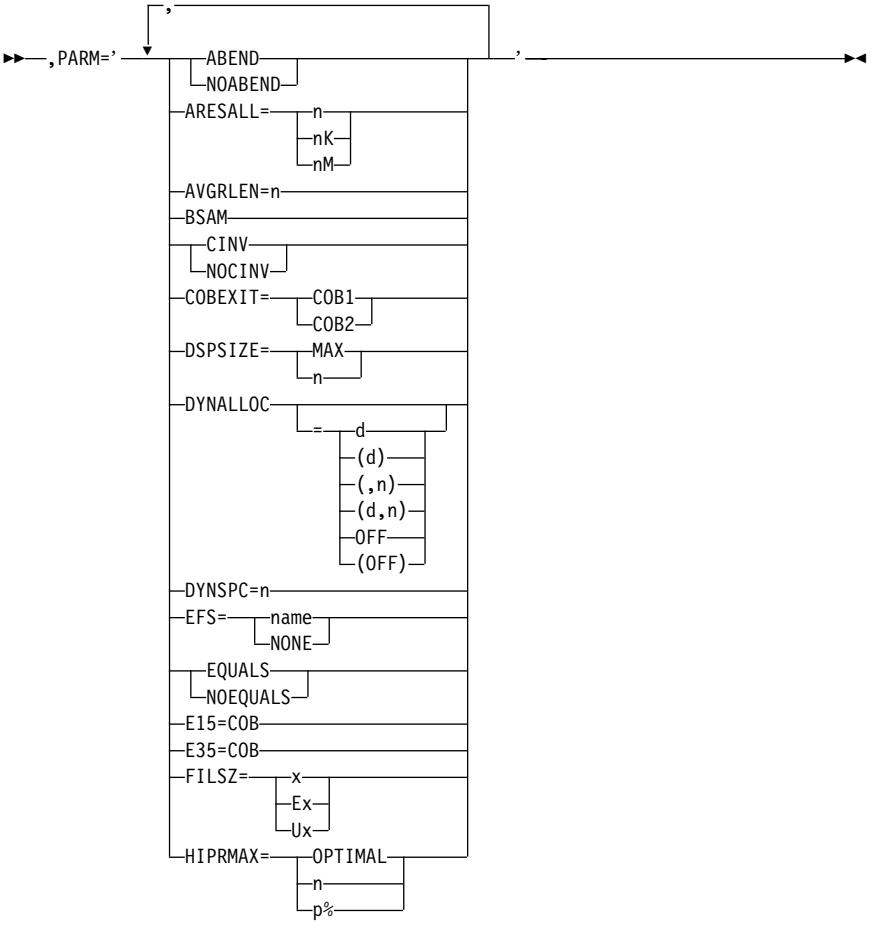



*Figure 4. DFSORT EXEC PARM Job Control Statement (Part 1 of 3)*

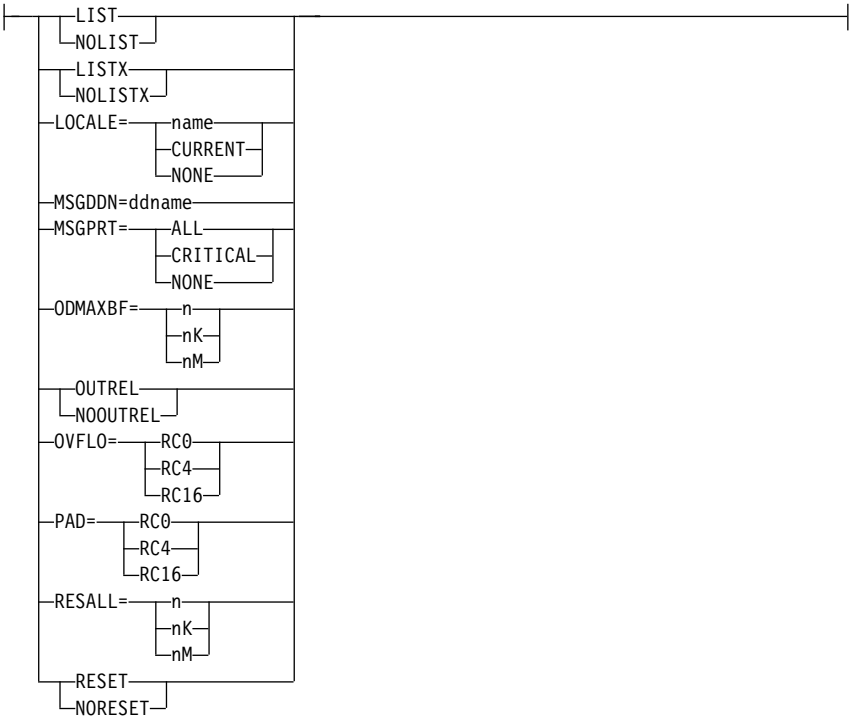



*Figure 4. DFSORT EXEC PARM Job Control Statement (Part 2 of 3)*

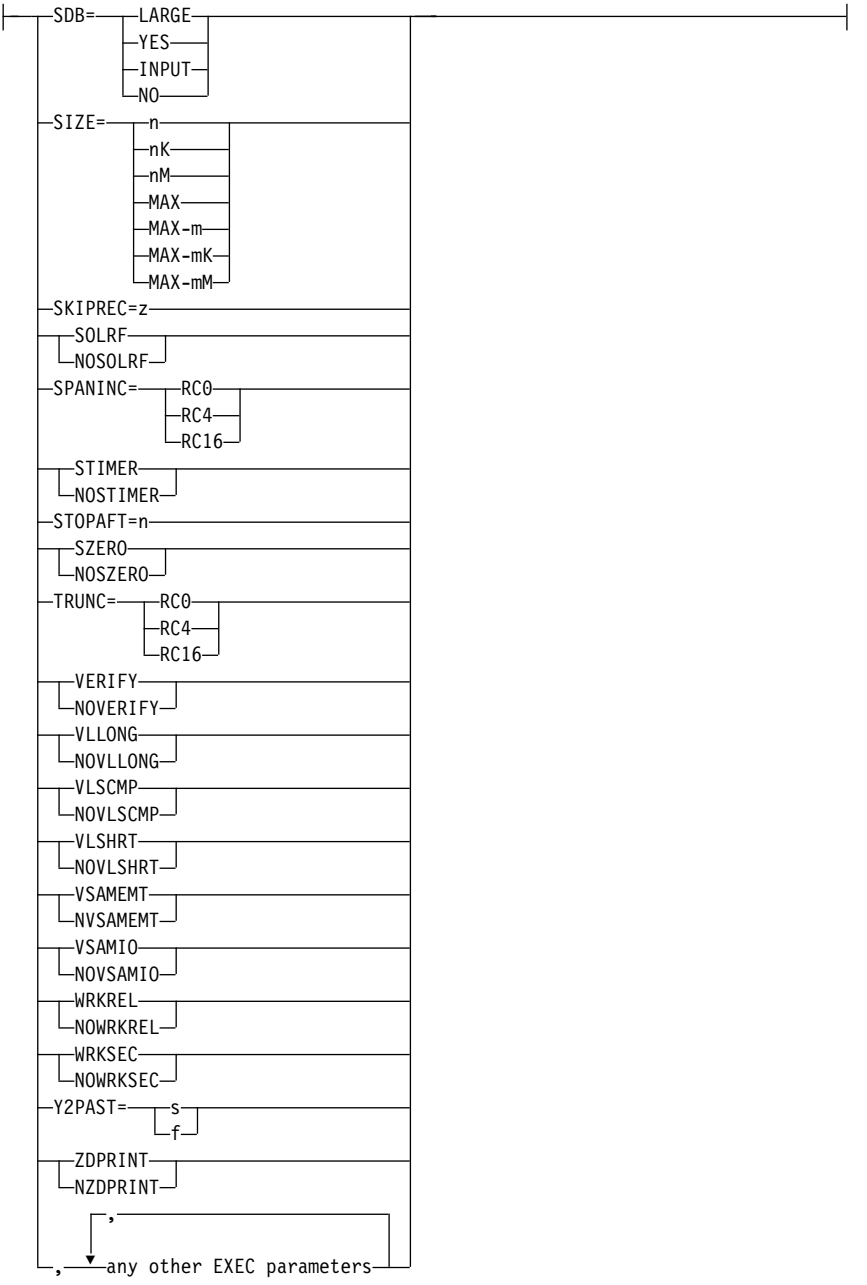



*Figure 4. DFSORT EXEC PARM Job Control Statement (Part 3 of 3)*

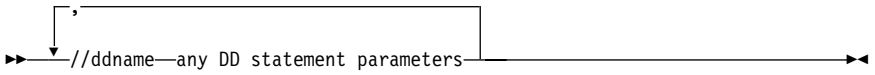



*Figure 5. //ddname Job Control Statement*

## JCL Description

**DD Statement**
**Description**

**//JOBLIB DD**
Defines your program link library if it is not already known to the system.

**//STEPLIB DD**
Same as //JOBLIB DD

**//SORTLIB DD**
Defines the data set that contains special load modules, if it is not already known to the system.

**//SYSOUT DD**
Defines the message data set.

**//SYMNAMES DD**
Defines the SYMNAMES data set containing statements to be used for symbol processing.

**//SYMNOUT DD**
Defines the data set in which SYMNAMES statements and the symbol table are to be listed.

**//SORTIN DD**
Defines the input data set for a sort or copy.

**//SORTINnn DD**
Defines the input data sets for a merge.

**//SORTOUT DD**
Defines the SORTOUT output data set for a sort, merge, or copy.

**//outfil DD**
Defines an OUTFIL data set for a sort, merge, or copy.

**//SORTWKdd DD**
Defines intermediate storage data sets for a sort.

**//DFSPARM DD**
Contains DFSORT PARM options and program control statements.

### JCL Description

**//SYSIN DD**
Contains DFSORT program control statements.

**//SORTCNTL DD**
Same as //SYSIN DD

**//SORTDIAG DD**
Specifies that all messages and program control statements be printed.

**//SORTCKPT DD**
Defines the data set for checkpoint records.

**//SYSUDUMP DD**
Defines the data set for output from a system ABEND dump routine.

**//SYSABEND DD**
Same as //SYSUDUMP DD

**//SYSMDUMP DD**
Same as //SYSUDUMP DD

**//SORTSNAP DD**
Defines the snap dump data set dynamically allocated by DFSORT.

**//ddname**
Defines the data set containing exit routines (as specified in the MODS program control statement).

The following DD statements are only necessary for dynamic link-editing of exit routines.

**//SYSPRINT DD**
Defines the message data set for the linkage editor.

**//SYSUT1 DD**
Defines the intermediate storage data set for the linkage editor.

**//SYSLIN DD**
Defines the data set for control information for the linkage editor.

**//SYSLMOD DD**
Defines the data set for output from the linkage editor.

**//SORTMODS DD**
Defines the temporary partitioned data set for exit routines from SYSIN.

## PARM=Options Definitions

| Option | Description |
|---|---|
| ABEND | Terminates an unsuccessful run with a user abend.<br>**Note:** RC16=ABE can be used instead of ABEND. |

| Option | Description |
|---|---|
| NOABEND | Terminates an unsuccessful run with a return code of 16.<br>**Note:** NORC16 can be used instead of NOABEND. |
| ARESALL= | Reserves main storage above 16-megabyte virtual for system and application use.<br>**n** Reserves n bytes of main storage.<br>**nK** Reserves n times K (K=1024) bytes of main storage.<br>**nM** Reserves n times M (M=1 048 576) bytes of main storage.<br>**Note:** RESERVEX can be used instead of ARESALL. |
| AVGRLEN= | Specifies the average input record length in bytes for variable-length record sort applications.<br>**Note:** L5=n can be used instead of AVGRLEN=n. |
| BSAM | Bypasses EXCP access method for input and output data sets. |
| CINV | Allows control interval access for VSAM data sets. |
| NOCINV | Does not allow control interval access for VSAM data sets. |
| COBEXIT= | Specifies the library for COBOL E15 and E35 routines.<br>**COB1** COBOL E15 and E35 routines run with the OS/VS COBOL run-time library or with no COBOL run-time library.<br>**COB2** COBOL E15 and E35 routines run with either the VS COBOL II run-time library or the Language Environment run-time library. |
| DSPSIZE= | Specifies the maximum amount of data space to be used with dataspace sorting.<br>**MAX** Specifies that DFSORT dynamically determines the maximum amount of data space to be used.<br>**n** Specifies the maximum amount, in megabytes, of data space to be used. |
| DYNALLOC | Allocates work data sets dynamically with values for d and n specified (or defaulted) by the DYNALOC option of the ICEMAC installation macro. |
| DYNALLOC= | Allocates work data sets dynamically.<br>**d** Specifies the device name.<br>**n** Specifies the maximum number of requested work data sets. |
| DYNALLOC=OFF | Prevents dynamic allocation of work data sets. |

## PARM=Options Definitions

| Option | Description |
|---|---|
| DYNSPC= | Specifies the total primary space for all of the dynamically allocated work data sets when the file size is unknown.<br>**n** Specifies the primary space in megabytes. |
| EFS= | Specifies the name of an EFS program for DFSORT to call.<br>**name** Calls the EFS program whose name you specify.<br>**NONE** Does not call an EFS program. |
| EQUALS | Specifies that the original sequence of records that collate identically must be preserved. |
| NOEQUALS | Specifies that the original sequence of records that collate identically need not be preserved. |
| E15=COB | Specifies that the E15 routine is written in COBOL. |
| E35=COB | Specifies that the E35 routine is written in COBOL. |
| FILSZ= | Specifies the number of records to be processed.<br>**x** The exact number of records to be processed.<br>**Ex** The estimated number of records to be processed. x must be immediately preceded by the letter E.<br>**Ux** The number of records to be processed. x must be immediately preceded by the letter U. |
| HIPRMAX= | Allocates Hiperspace for Hipersorting.<br>**OPTIMAL** Determines dynamically the maximum amount of Hiperspace.<br>**n** Determines dynamically the maximum amount of Hiperspace up to n megabytes.<br>**p%** Determines dynamically the maximum amount of Hiperspace up to p percent of configured expanded storage (31-bit mode) or of a portion of central storage (64-bit mode). |
| LIST | Prints DFSORT control statements. |
| NOLIST | Does not print DFSORT control statements. |
| LISTX | Prints control statements returned by an EFS program. |
| NOLISTX | Does not print control statements returned by an EFS program. |

| Option | Description |
|---|---|
| LOCALE= | Designates active locale.<br>**name** Name of active locale.<br>**CURRENT** Current locale remains active.<br>**NONE** Does not use locale processing. |
| MSGDDN= | Specifies the message data set ddname.<br>**Note:** MSGDD can be used instead of MSGDDN. |
| MSGPRT= | Specifies the message type.<br>**ALL** Prints all messages, except diagnostic messages.<br>**NONE** Does not print messages and control statements.<br>**CRITICAL** Prints only critical (error) messages.<br>**Note:** FLAG(I), FLAG(U), NOFLAG, or MSG=xx can be used instead of MSGPRT. |
| ODMAXBF= | Specifies maximum amount of OUTFIL data set buffer space.<br>**n** Specifies n bytes of buffer space.<br>**nK** Specifies n times K (K=1024) bytes of buffer space.<br>**nM** Specifies n times M (M=1 048 576) bytes of buffer space. |
| OUTREL | Releases unused temporary output data set space.<br>**Note:** RLSOUT can be used instead of OUTREL. |
| NOOUTREL | Does not release unused temporary output data set space.<br>**Note:** NORLSOUT can be used instead of NOOUTREL. |
| OVFLO= | Specifies the action to be taken by DFSORT when BI, FI, PD or ZD summary fields overflow.<br>**RC0** Continues processing and sets a return code of 0.<br>**RC4** Continues processing and sets a return code of 4.<br>**RC16** Terminates and gives a return code of 16. |

## PARM=Options Definitions

| Option | Description |
|---|---|
| PAD= | Specifies the action to be taken by DFSORT for LRECL padding.<br>**RC0** Continues processing and sets a return code of 0.<br>**RC4** Continues processing and sets a return code of 4.<br>**RC16** Terminates and gives a return code of 16. |
| RESALL= | Reserves main storage below 16-megabyte virtual for system and application use when SIZE/MAINSIZE=MAX is in effect.<br>**n** Reserves n bytes of main storage.<br>**nK** Reserves n times K (K=1024) bytes of main storage.<br>**nM** Reserves n times M (M=1 048 576) bytes of main storage.<br>**Note:** RESERVE can be used instead of RESALL. |
| RESET | Processes a VSAM output data set with REUSE as NEW. |
| NORESET | Processes a VSAM output data set with REUSE as MOD. |
| SDB= | Specifies whether DFSORT should use the system-determined optimum block size for output data sets when the block size is specified as zero or defaulted to zero.<br>**LARGE** Uses the system-determined optimum block size for output data sets. Allows DFSORT to select a block size greater than 32760 bytes for a tape output data set, when appropriate..<br>**YES** Uses the system-determined optimum block size for output data sets, but limits the selected block size to a maximum of 32760 bytes. **Note:** SDB, SDB=ON and SDB=SMALL can be used instead of SDB=YES..<br>**INPUT** Uses the system-determined optimum block size for output data sets, but limits the selected block size to a maximum of 32760 bytes if the input block size is less than or equal to 32760 bytes.<br>**NO** Does not use the system-determined optimum block size for output data sets. **Note:** NOSDB and SDB=OFF can be used instead of SDB=NO. |

| Option | Description |
|---|---|
| SIZE= | Specifies amount of main storage available:<br>**n** Allocates n bytes of main storage.<br>**nK** Allocates n times K (K=1024) bytes of main storage.<br>**nM** Allocates n times M (M=1 048 576) bytes of main storage.<br>**MAX** Specifies main storage to be allocated as indicated by MAXLIM or TMAXLIM installation option.<br>**MAX-m** Allocates main storage specified by MAXLIM or TMAXLIM and the RESALL value (m). Specifying SIZE=MAX-m is equivalent to specifying SIZE=MAX and RESALL=m.<br>**MAX-mK** Allocates main storage specified by MAXLIM or TMAXLIM and the RESALL value (mK). Specifying SIZE=MAX-mK is equivalent to specifying SIZE=MAX and RESALL=mK.<br>**MAX-mM** Allocates main storage specified by MAXLIM or TMAXLIM and the RESALL value (mM). Specifying SIZE=MAX-mM is equivalent to specifying SIZE=MAX and RESALL=mM.<br>**Note:** SIZE(option), CORE=option, and CORE(option) can be used instead of SIZE=option. |
| SKIPREC= | Skips z input records before DFSORT begins to sort or copy. |
| SOLRF | Uses the reformatted record length for the SORTOUT LRECL.<br>**Note:** CAOUTREC can be used instead of SOLRF. |
| NOSOLRF | Does not use the reformatted record length for the SORTOUT LRECL. |
| SPANINC= | Specifies the action to be taken by DFSORT when incomplete spanned records are detected.<br>**RC0** Continues processing and sets a return code of 0.<br>**RC4** Continues processing and sets a return code of 4.<br>**RC16** Terminates and gives a return code of 16. |
| STIMER | Uses the STIMER macro. |

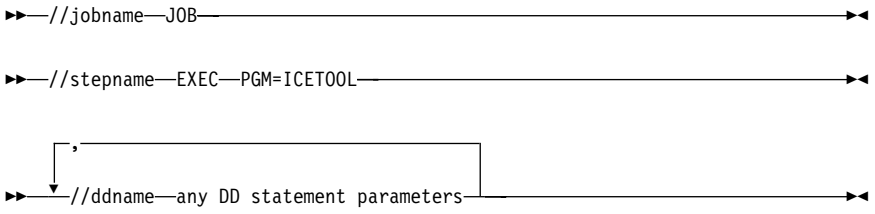
## PARM=Options Definitions

| Option | Description |
|---|---|
| NOSTIMER | Does not use the STIMER macro. |
| STOPAFT= | Indicates the number of records you want accepted for sorting or copying. |
| SZERO | Treats − 0 and + 0 as signed, that is, as the same. |
| NOSZERO | Treats − 0 and + 0 as unsigned, that is, as different. |
| TRUNC= | Specifies the action to be taken by DFSORT for LRECL truncation.<br>**RC0** Continues processing and sets a return code of 0.<br>**RC4** Continues processing and sets a return code of 4.<br>**RC16** Terminates and gives a return code of 16. |
| VERIFY | Verifies the sequence of the final output records.<br>**Note:** SEQ=YES can be used instead of VERIFY. |
| NOVERIFY | Does not verify the sequence of the final output records.<br>**Note:** SEQ=NO can be used instead of NOVERIFY. |
| VLLONG | Truncates long variable-length output records. |
| NOVLLONG | Stops processing if a long variable-length output record is found. |
| VLSCMP | Pads short variable-length compare fields with binary zeros. |
| NOVLSCMP | Does not pad short variable-length compare fields. |
| VLSHRT | Continues processing if a short variable-length control field, compare field, or summary field is found. |
| NOVLSHRT | Stops processing if a short variable-length control field, compare field, or summary field is found. |
| VSAMEMT | Processes an empty VSAM input data set. |
| NVSAMEMT | Terminates for an empty VSAM input data set. |
| VSAMIO | Allows same VSAM data set with REUSE to be sorted in-place. |
| NOVSAMIO | Terminates if same VSAM data set used for input and output. |
| WRKREL | Releases unused temporary SORTWKdd data set space.<br>**Note:** RELEASE=ON and RLS=n (n greater than 0) can be used instead of WRKREL. |
| NOWRKREL | Does not release unused temporary SORTWKdd data set space.<br>**Note:** RELEASE=OFF and RLS=0 can be used instead of NOWRKREL. |

| Option | Description |
|---|---|
| WRKSEC | Uses automatic secondary allocation for temporary SORTWKdd data sets. 25% of the primary allocation is used.<br>**Note:** SECOND=ON and SEC=n (n greater than 0) can be used instead of WRKSEC. |
| NOWRKSEC | Does not use automatic secondary allocation for temporary SORTWKdd data sets.<br>**Note:** SECOND=OFF and SEC=0 can be used instead of NOWRKSEC. |
| Y2PAST= | Specifies the sliding or fixed century window.<br>**s** Starts the sliding century window at the current year − s.<br>**f** Starts the fixed century window at f.<br>**Note:** CENTWIN can be used instead of Y2PAST. |
| ZDPRINT | Converts positive ZD summation results to printable numbers.<br>**Note:** ZDPRINT=YES can be used instead of ZDPRINT. |
| NZDPRINT | Does not convert positive ZD summation results to printable numbers.<br>**Note:** ZDPRINT=NO can be used instead of NZDPRINT. |

# Chapter 5. ICETOOL Job Control Statements

Details on coding ICETOOL job control language (JCL) statements are explained in *Application Programming Guide*




## JCL Description

**DD Statement**
**Description**

**//JOBLIB DD**
Defines your program link library if it is not already known to the system.

**//STEPLIB DD**
Same as //JOBLIB DD.

**//TOOLMSG DD**
Defines the ICETOOL message data set for all operations.

**//DFSMSG DD**
Defines the DFSORT message data set for all operations.

**//SYMNAMES DD**
Defines the SYMNAMES data set containing statements to be used for symbol processing.

**//SYMNOUT DD**
Defines the data set in which SYMNAMES statements and the symbol table are to be listed.

**//TOOLIN DD**
Contains ICETOOL operator and comment statements.

**//indd DD**
Defines an input data set for a COPY, COUNT, DISPLAY, OCCUR, RANGE, SELECT, SORT, STATS, UNIQUE, or VERIFY operation.

**//outdd DD**
Defines an output data set for a COPY, SELECT, or SORT operation.

**//savedd DD**
Defines an output data set for a SELECT operation.

**//xxxxCNTL DD**
Contains DFSORT control statements for a COPY, COUNT, or SORT operation.

# Chapter 6. Notices

References in this publication to IBM® products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights or other legally protectible rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594.

## Trademarks

The following terms are trademarks or service marks of the IBM Corporation in the United States, or other countries, or both:

DFSORT
Hiperspace
Hipersorting
IBM
MVS/ESA
OS/390
z/OS

Other company, product, and service names may be trademarks or service marks of other companies.

# Readers' Comments — We'd Like to Hear from You

**DFSORT**
**Reference Summary**
**Release 14**

**Publication No. SX33-8001-14**

**Overall, how satisfied are you with the information in this book?**

| | Very Satisfied | Satisfied | Neutral | Dissatisfied | Very Dissatisfied |
|---|---|---|---|---|---|
| Overall satisfaction | ☐ | ☐ | ☐ | ☐ | ☐ |

**How satisfied are you that the information in this book is:**

| | Very Satisfied | Satisfied | Neutral | Dissatisfied | Very Dissatisfied |
|---|---|---|---|---|---|
| Accurate | ☐ | ☐ | ☐ | ☐ | ☐ |
| Complete | ☐ | ☐ | ☐ | ☐ | ☐ |
| Easy to find | ☐ | ☐ | ☐ | ☐ | ☐ |
| Easy to understand | ☐ | ☐ | ☐ | ☐ | ☐ |
| Well organized | ☐ | ☐ | ☐ | ☐ | ☐ |
| Applicable to your tasks | ☐ | ☐ | ☐ | ☐ | ☐ |

**Please tell us how we can improve this book:**

Thank you for your responses. May we contact you? ☐ Yes ☐ No

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you.

Name

Address

Company or Organization

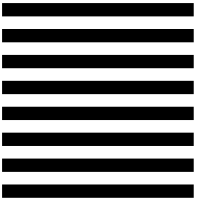
Phone No.

**Readers' Comments — We'd Like to Hear from You**
SX33-8001-14

Cut or Fold Along Line

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation
RCF Processing Department
M86/050
5600 Cottle Road
SAN JOSE, CA 95193-0001

Fold and Tape **Please do not staple** Fold and Tape

SX33-8001-14

Cut or Fold Along Line