DFSORT/VSE V3R4: ICETOOL Mini-User Guide

May, 1998

Frank L. Yaeger

DFSORT Team
IBM Storage Systems Division
San Jose, California
VNET: YAEGER at SJFEVMX
IBMMAIL: USIB257Z at IBMMAIL
Internet: fyaeger@vnet.ibm.com

DFSORT/VSE Web Site

Come visit DFSORT/VSE on the World Wide Web at URL:

http://www.ibm.com/storage/dfsortvse/

Abstract

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

Contents

ICETOOL Mini-User Guide

Introduction

ICETOOL, a versatile file processing and reporting utility, provides an easy-to-use batch front-end for DFSORT/VSE. ICETOOL combines new features with previously available DFSORT/VSE features to perform complex sorting, copying, reporting and analytical tasks using multiple files in a single job step.

This document is a mini-user guide for the DFSORT/VSE V3R4 version of ICETOOL. The major features of ICETOOL, including its JCL and control statements, are discussed at length using many examples. The objective is to show you how to use ICETOOL to accomplish complex tasks.

Complete information on ICETOOL′s JCL, control statements, restrictions, calling program interface, messages and return codes can be found in ″DFSORT/VSE Application Programming Guide″ (SC26-7040).

Additional Sources for ICETOOL Examples

- ″DFSORT/VSE Application Programming Guide″ (SC26-7040)
- ″Getting Started with DFSORT/VSE″ (SC26-7101)

What Can ICETOOL Do?

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

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

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

- Printing statistical information for selected numeric fields, such as minimum, maximum, average, total, count of values within a range and count of unique values. This makes it easy to extract frequently used analytical data.
- Printing character and numeric fields in a variety of report formats, allowing control of title, date, time, page numbers, headings, lines per page, field formats, and total, maximum, minimum and average values for the columns of numeric data. This makes it easy to create simple, tailored and sectioned reports.
- Identifying and printing invalid decimal values and their locations in a file. This makes it easy to avoid using invalid fields for other operations.
- Creating multiple copies of sorted, edited, or unedited files. This makes it easy to create several identical file copies.
- Creating output files containing different subsets or field arrangements of input files. This makes it easy to view data in many different ways.
- Printing the DFSORT/VSE installation defaults selected at your site. This makes it easy to determine the options selected and accepted.

- Creating output files, and printing reports in a variety of formats, for records with duplicate values, non-duplicate values, or values that occur n times, less than n times, or more than n times. This makes it easy to view data according to occurrences of values.
- Printing reports in a variety of formats, showing unique values for selected character and numeric fields and the number of times each occurs. This makes it easy to obtain reports based on occurrences of values.
- Allowing operations to be performed or suppressed based on the success or failure of previous operations. This makes it easy to group operations according to the action to be taken after an error.

ICETOOL can be called directly or from a program. When ICETOOL is called directly, ICETOOL operator statements and DFSORT/VSE control statements (if needed), must be supplied in the SYSIPT file. When ICETOOL is called from a program, ICETOOL operator statements and DFSORT/VSE control statements (if needed), must be supplied in a calling program parameter list, in which case ICETOOL returns information for each operation in the parameter list. In either case, ICETOOL prints messages and gives a return code for each operation.

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

General ICETOOL Job Format

Here's a representation of an ICETOOL job:

```
// JOB EXAMP ...
// ASSGN SYS012,SYSLST
// ASSGN SYS013,01E
 .
 .
 .
// DLBL infile1
// TLBL infile2
// DLBL outfile1
// TLBL outfile2
// DLBL sortwk1
 .
 .
 .
// EXEC ICETOOL,SIZE=100K
 <ICETOOL statements and DFSORT/VSE control statements>
/*
/&
```

JCL Statements

// ASSGN SYSnnn,SYSLST
Defines a printer for ICETOOL reports or DFSORT/VSE messages.

// ASSGN SYSnnn,cuu
Defines a printer address (cuu) for routing DFSORT/VSE messages or ICETOOL reports to a specific printer. VSE/POWER statement LST can be used instead for spooling DFSORT/VSE messages or ICETOOL reports to a specific printer queue.

Note: A printer with the specified cuu address must have been defined to the VSE/ESA operating system.

DLBL infile1
Defines an input disk file for a COPY, COUNT, DISPLAY, OCCUR, RANGE, SELECT, SORT, STATS, UNIQUE or VERIFY operator.

TLBL infile2
Defines an input tape file for a COPY, COUNT, DISPLAY, OCCUR, RANGE, SELECT, SORT, STATS, UNIQUE or VERIFY operator.

DLBL outfile1
Defines an output disk file for a COPY, SELECT or SORT operation.

TLBL outfile2
Defines an output tape file for a COPY, SELECT or SORT operation.

DLBL sortwk1
Defines intermediate storage for DFSORT/VSE for an OCCUR, SELECT, SORT or UNIQUE operation.

The SYSIPT file is used for ICETOOL statements and DFSORT/VSE control statements. SYSLST is used for ICETOOL messages.

You must supply JCL statements for all of the input and output files needed for the ICETOOL operations you specify. In addition, you must supply all of the JCL statements needed for a DFSORT/VSE application.

ICETOOL Operator Statement Syntax

Operator Statements

- Each ICETOOL operator statement describes a task you want ICETOOL to perform.
- With the exception of DEFINE, any number of operators can be specified and in any order. Up to 50 DEFINE operators with unique file names can be specified in an ICETOOL run and each DEFINE operator must precede all operators it is associated with.
- The general format for all ICETOOL statements is:

  operator operand ... operand

  Example:

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

  Example:

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

DFSORT/VSE Section

The SORT operator requires a DFSORT/VSE SORT control statement to indicate the fields to be sorted. The SORT, COPY and COUNT operators all allow DFSORT/VSE control statements such as INCLUDE, OMIT, OUTREC and OPTION to be specified so that you can create subsets, rearrange the fields of your records, and so on.

A DFSORT/VSE section is used to supply the DFSORT/VSE control statements for a SORT, COPY or COUNT operator. Here′s an example of a SORT operator with a DFSORT/VSE section:

```
SORT FROM(IN1) TO(OUT1) USE
USTART
 SORT FIELDS=(5,4,CH,A,22,3,PD,D)
 INCLUDE COND=(32,3,CH,EQ,C′L92′,OR,
               32,3,CH,EQ,C′J69′)
 OPTION DSPSIZE=1
UEND
```

The USE operand indicates that a DFSORT/VSE section follows the operator statement. The DFSORT/VSE section must consist of a USTART delimiter, the DFSORT/VSE control statements and a UEND delimiter. A DFSORT/VSE section can contain comment statements and blank statements.

Comment Statements

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

Blank Statements

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

ICETOOL Return Codes

ICETOOL sets a return code for each operation it performs.

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

The return codes are:

- 0 - Successful completion. No errors were detected.
- 4 - Successful completion. DFSORT/VSE detected a warning condition.
- 12 - Unsuccessful completion. ICETOOL detected one or more errors. Messages for these errors were printed on SYSLST.
- 16 - Unsuccessful completion. DFSORT/VSE detected one or more errors. Critical error messages for these errors were routed to the system console. If the ROUTE=xxx operand was specified, all DFSORT/VSE messages were routed to SYSxxx.
- 20 - Unsuccessful completion. ICETOOL could not obtain 12 kilobytes from the 24-bit GETVIS area.
- 24 - Unsuccessful completion. ICETOOL could not load the phase required for ICETOOL operations. The message for this error was printed on SYSLST.

ICETOOL Job with all Operators

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

```
// JOB EXAMP
// ASSGN SYS015,SYSLST
// ASSGN SYS016,SYSLST
// ASSGN SYS001,X′E4F′
// ASSGN SYS002,X′E4D′
// DLBL IN1,′FLY.INPUT1′,,VSAM,DISP=(OLD,KEEP)
// DLBL IN2,′FLY.INPUT2′,,VSAM,DISP=(OLD,KEEP)
// DLBL IN3,′FLY.INPUT3′,,VSAM,DISP=(OLD,KEEP)
// DLBL VLRIN,′FLY.VLR′,,SD
// EXTENT SYS002,339001,,,900,200
// DLBL DASD,′FLY.COPY′,,SD
// EXTENT SYS001,339000,,,1900,500
// ASSGN SYS012,181
// TLBL TAPE,′FLY.BACKUP′,,222222,,1
// DLBL SEL1,′FLY.SELECT′,,VSAM,DISP=(NEW,KEEP),RECORDS=100,RECSIZE=100
// EXTENT ,SYSWK1
// DLBL OUT1,′FLY.SORT′,,SD
// EXTENT SYS001,339000,,,2500,500
// EXEC ICETOOL,SIZE=100K
* ICETOOL Operators - brief explanations and simple examples

* DEFINE - defines file characteristics as needed.
  DEFINE NAME(IN1) TYPE(F) LENGTH(100)
  DEFINE NAME(IN2) TYPE(F) LENGTH(120)
  DEFINE NAME(IN3) TYPE(F) LENGTH(100)
  DEFINE NAME(VLRIN) TYPE(V) LENGTH(85)
  DEFINE NAME(TAPE) UNIT(012)
  DEFINE NAME(OUT1) TYPE(F) LENGTH(100)

* COPY - copies one or more input files to one or more
* output files.
* Example: copy the IN1 and IN2 files to the DASD and
* TAPE files.
  COPY FROM(IN1,IN2) TO(DASD,TAPE)

* COUNT - prints a message containing the count of records
* in one or more input files.
* Example: print a count of the number of records in the
* IN2 file.
  COUNT FROM(IN2)

* DEFAULTS - prints the DFSORT/VSE installation defaults.
* Example: print the defaults to SYSLST.
  DEFAULTS LIST(LST)
```

```
* DISPLAY - prints the values and characters of specific
* numeric and character fields.  Simple, tailored or
* sectioned reports can be produced.
* Example: print a tailored report showing values from the
* three IN2 file ON fields, with title, heading lines,
* maximum and minimum, on the printer associated with
* logical unit SYS015.
  DISPLAY FROM(IN2) LIST(015) BLANK -
  TITLE('IN2 File Report') PAGE DATE TIME -
  HEADER('Store') ON(50,15,CH) -
  HEADER('Profit/(Loss)') ON(26,8,ZD,E1) -
  HEADER('Employees') ON(2,3,ZD) -
  MAXIMUM('Largest') MINIMUM('Smallest')

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

* OCCUR - prints each unique value for specified numeric and
* character fields and how many times it occurs.
* Simple or tailored reports can be produced.
* The values printed can be limited to those for which the
* value meets specified criteria (e.g. only duplicate values).
* Example: print a report showing values from the IN1 file
* ON field, with the number of times each value occurs, on the
* printer associated with SYS016.
  OCCUR FROM(IN1) LIST(016) TITLE('Dep''t Counts') -
    ON(35,3,CH) ON(VALCNT)

* RANGE - prints a message containing the count of values in
* a specified range for a specified numeric field in one or
* more input files.
* Example: print a count of the values in the IN2 file
* ON field that are higher than -50, but lower than +100.
  RANGE FROM(IN2) ON(20,2,PD) HIGHER(-50) LOWER(100)

* SELECT - selects records from one or more input files for
* inclusion in an output file based on meeting criteria for
* the number of times specified numeric or character field
* values occur (e.g. only duplicate values).
* Example: selects records from the IN1 file, for the SEL1 file,
* whose ON field occurs only once (i.e, only records with
* no duplicate ON field values).
  SELECT FROM(IN1) TO(SEL1) ON(30,2,PD) NODUPS
```

```
* SORT - sorts one or more input files to one or more
* output files.
* Example: sort the IN1 and IN3 files to the OUT1 file using
* the SORT and OMIT statements in the DFSORT/VSE section.
* For variable-length records, ON(VLEN) gives statistics about
* the length of the records.
  SORT FROM(IN1,IN3) TO(OUT1) USE
  USTART
   SORT FIELDS=(35,3,CH,A)
   OMIT COND=(5,6,CH,EQ,C'DELETE')
  UEND

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

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

* VERIFY - examines specified decimal fields in one or more
* input files and prints a message identifying each invalid
* value found for each field.
* Example: identify all values in the two IN2 file
* decimal ON fields that have invalid digits (A-F)
* and/or invalid signs (0-9).
  VERIFY FROM(IN2) ON(10,2,ZD) ON(41,6,PD)
/*
/&
```

COPY Operator Details

Syntax

```
COPY FROM(filename,...) TO(filename,...) USE LOCALE(name)
                                             LOCALE(CURRENT)
                                             LOCALE(NONE)
```

Function

Copies the input files to the output files.

The input files are copied to the first output file. The DFSORT/VSE control statements in the DFSORT/VSE section are used if USE is specified. DFSORT/VSE control statements can be used to copy a subset of the input records (INCLUDE or OMIT statement), reformat records for output (OUTREC statement), and so on. See ″DFSORT/VSE Application Programming Guide″ for complete details of DFSORT/VSE control statements.

If the first copy is successful, the first output file is copied to the second and subsequent output files (if any).

Examples

```
COPY FROM(ADATA,BDATA) TO(OUT1,OUT2,OUT3) USE
USTART
  INCLUDE COND=(21,8,ZD,GT,50000)
UEND
COPY FROM(VSAMIN) TO(VSAMOUT)
COPY FROM(INPUT) TO(DASD,TAPE1,TAPE2)
```

Required Operands

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.
- TO - the file names of 1 to 10 output files. You must supply JCL statements for the output files you specify. You must supply DEFINE operators for tape output files and VSAM output files you specify.

Optional Operands

- USE - indicates that DFSORT/VSE control statements in the DFSORT/VSE section that appears immediately after this COPY statement are to be used for this operation.
- LOCALE - overrides the installation default for locale processing.

Notes

- If more than one output file is specified, DFSORT/VSE must be able to read the first output file after it is written in order to copy it to the other output files.
- For maximum efficiency, use a DASD file as the first in a list of output files on both DASD and tape since it is more efficient to write and then read a DASD file than a tape file.

Example of Message Output for COPY

```
      * COPY - copies one or more input files to one or more
      * output files.
      * Example: copy the IN1 and IN2 files to the DASD and
      * TAPE files.
        COPY FROM(IN1,IN2) TO(DASD,TAPE)
7T27I DFSORT/VSE CALL 0001 FOR COPY FROM IN1     TO DASD     COMPLETED
7T27I DFSORT/VSE CALL 0002 FOR COPY FROM DASD    TO TAPE     COMPLETED
7T02I OPERATION RETURN CODE:  00
```

COUNT Operator Details

Syntax

```
COUNT FROM(filename,...) USE LOCALE(name)
                             LOCALE(CURRENT)
                             LOCALE(NONE)
```

Function

Prints a message containing the count of records in the input files. The DFSORT/VSE control statements in the DFSORT/VSE section are used if USE is specified. The INCLUDE or OMIT statement can be used to count a subset of the input records.

Examples

```
COUNT FROM(IN1)
COUNT FROM(IN2,IN3) USE
USTART
OMIT COND=(15,2,ZD,GT,32,
     AND,28,2,CH,EQ,C′CA′)
UEND
```

Required Operand

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.

Optional Operands

- USE - indicates that DFSORT/VSE control statements in the DFSORT/VSE section that appears immediately after this COUNT statement are to be used for this operation.
- LOCALE - overrides the installation default for locale processing.

Note

- The record count is also printed for the DISPLAY, OCCUR, RANGE, SELECT, STATS, UNIQUE, and VERIFY operators.

Example of Message Output for COUNT

```
      * COUNT - prints a message containing the count of records
      * in one or more input files.
      * Example: print a count of the number of records in the
      * IN2 file.
        COUNT FROM(IN2)
7T27I DFSORT/VSE CALL 0003 FOR COPY FROM IN2      TO E35 EXIT COMPLETED
7T28I RECORD COUNT:  000000000000008
7T02I OPERATION RETURN CODE:  00
```

DEFAULTS Operator Details

Syntax

```
DEFAULTS LIST(LST)
         LIST(xxx)
```

Function

Prints the DFSORT/VSE installation (ILUINST) defaults. The value selected or accepted for each installation parameter is shown, along with the IBM-supplied value if a different value was selected.

Example

```
DEFAULTS LIST(011)
```

Required Operand

- LIST - where the output will be routed (LST for SYSLST or xxx for the printer associated with logical unit SYSxxx).

Notes

- The control character occupies the first byte of each record.
- ICETOOL uses CTLCHR=ASA and BLKSIZE=121 to define the file for the printer.

Example of Message Output for DEFAULTS

```
      * DEFAULTS - prints the DFSORT/VSE installation defaults.
      * Example: print the defaults to SYSLST.
        DEFAULTS LIST(LST)
7T03I INFORMATION PRINTED ON SYSLST
7T02I OPERATION RETURN CODE:  00
```

Example of Report Output for DEFAULTS

```
DFSORT/VSE INSTALLATION (ILUINST) DEFAULTS               - 1 -

* ONLY SHOWN IF DIFFERENT FROM THE SPECIFIED INSTALLATION DEFAULT

PARAMETER     INSTALLATION DEFAULT      IBM-SUPPLIED DEFAULT *
----------    ----------------------    ----------------------
CHALT         NOCHALT
DIAG          DIAG                      NODIAG
DUMP          NODUMP
EQUALS        NOEQUALS
ERASE         NOERASE
 .
 .
 .
```

DEFINE Operator Details (file characteristics)

Syntax

```
DEFINE NAME(filename) TYPE(x) LENGTH(n) BLOCKSIZE(n)
                                        BLKSIZE(n)

       VSAMIN VSAMOUT(yyyy) TOLERATE REUSE UNIT(xxx)
                            TOL
```

Function

Supplies needed file characteristics for the input and output files used for COPY, COUNT, DISPLAY, OCCUR, RANGE, SELECT, SORT, STATS, UNIQUE and VERIFY operations. The DEFINE operator for a file must precede any operators which use that file.

Examples

```
DEFINE NAME(INPUT) TYPE(F) LENGTH(173)
DEFINE NAME(TAPE) UNIT(011)
```

Required Operand

- NAME - the file name of an input file used in the FROM operand of a subsequent operator, or of an output file used in the TO operand of a subsequent operator.

Optional Operands

- TYPE - the input record format (F or V). TYPE must be specified for each file specified in a FROM operand.
- LENGTH - the input record length (TYPE=F) or maximum input record length (TYPE=V). LENGTH must be specified for each file specified in a FROM operand.
- BLOCKSIZE and BLKSIZE - the maximum input or output block size.
- VSAMIN - indicates the input file is VSAM. VSAMIN must be specified for each VSAM file specified in a FROM operand.
- VSAMOUT - indicates the output file is VSAM and gives its type (ESDS, RRDS or KSDS). VSAMOUT must be specified for each VSAM file specified in a TO operand.
- TOLERATE and TOL - indicates DFSORT/VSE should tolerate a warning when opening this VSAM input or output file.
- REUSE - indicates you want to write over this existing non-empty VSAM output file defined with the REUSE attribute.
- UNIT - indicates the logical unit number (001 to 221 for SYS001 to SYS221, respectively) for this input or output tape file. UNIT must be specified for each tape file specified in a FROM or TO operand.

Note

- Up to 50 DEFINE operators with unique file names can be specified for one ICETOOL run.

Example of Message Output for DEFINE

```
      * DEFINE - defines file characteristics as needed.
        DEFINE NAME(IN1) TYPE(F) LENGTH(100)
7T02I OPERATION RETURN CODE:  00

        DEFINE NAME(IN2) TYPE(F) LENGTH(120)
7T02I OPERATION RETURN CODE:  00

 .
 .
 .
```

DEFINE Operator Details (DFSORT/VSE message routing)

Syntax

```
DEFINE ROUTE(LST)
       ROUTE(LOG)
       ROUTE(xxx)
       ROUTE(CRI)
```

Function

Indicates where the DFSORT/VSE messages are routed.

Example

```
DEFINE ROUTE(012)
```

Required Operand

- ROUTE(LST), ROUTE(LOG), ROUTE(xxx) or ROUTE(CRI)
  - LST - informational messages are routed to SYSLST. Critical messages are routed to SYSLST and the system console.
  - LOG - informational and critical messages are routed to the system console.
  - xxx - informational and critical messages are routed to SYSxxx.
  - CRI - critical messages are routed to the system console. Informational messages are not issued.

Note

- If a DEFINE operator with ROUTE is not specified, ROUTE(CRI) is used.

DISPLAY Operator Details

Syntax

```
DISPLAY FROM(filename,...) LIST(LST) ON(p,m,f)           ...
                           LIST(xxx) ON(p,m,f,formatting)
                                     ON(p,m,HEX)
                                     ON(VLEN)
                                     ON(NUM)

        TITLE('string') PAGE DATE       TIME
                             DATE(abcd) TIME(abc)

        HEADER('string') ... LINES(n) BLANK
        HEADER(NONE)                  PLUS
        NOHEADER

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

        AVERAGE('string') LIMIT(n)

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

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

Function

Prints the values or characters of specified numeric and/or character fields. The fields are printed in columns in the same order in which they are specified. From 1 to 20 ON fields can be specified as long as the resulting line length does not exceed 121 bytes. Specifying the BLANK or PLUS operand can compress the columns of data allowing more fields in the report (up to a maximum of 20).

Simple, tailored or sectioned reports can be produced:

- A title line consisting of a string, the page number, the date and/or the time, in any order specified, can appear at the top of each page.
- Headings can be printed to identify each specified field.
- Overall statistics (total, maximum, minimum, and/or average) can be produced for numeric fields.
- Column widths are adjusted automatically according to the operands specified and the length of the fields.
- Alternate formats for date and time can be specified.
- Numeric fields can be edited with separators, decimal points, division, leading, trailing and floating signs.
- A break field can be used to produce sections with their own break title and break statistics (total, maximum, minimum and/or average).

Examples

```
DISPLAY FROM(DATA) LIST(LST) ON(10,44,CH) ON(5,4,FI)

DISPLAY FROM(VLR) LIST(016) ON(NUM) ON(VLEN) ON(1,4,HEX) -
  TITLE(′Record Length Report′) DATE -
  TOTAL(′Total Length′) AVERAGE(′Average Length′) -
  MINIMUM(′Minimum Length′) MAXIMUM(′Maximum Length′)

DISPLAY FROM(MASTER) LIST(015) LINES(50) -
        PAGE TITLE(′Report for Denmark′) DATE(DM4-) -
        BTITLE(′Division:′) BREAK(23,10,CH) -
        HEADER(′Part′) ON(15,6,CH) -
        HEADER(′Completed′) ON(3,4,ZD,A2) -
        HEADER(′Value (kr)′) ON(38,8,ZD,C2) -
        BMINIMUM(′Lowest in this Division:′) -
        BMAXIMUM(′Highest in this Division:′) -
        BAVERAGE(′Average in this Division:′) -
        BTOTAL(′Total for this Division:′) -
        AVERAGE(′Average for all Divisions:′)

DISPLAY FROM(DATA1,DATA2) LIST(LST) -
   NOHEADER ON(15,44,CH) ON(8,4,PD,/KB,T′ KB′)
```

Required Operands

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.

- LIST - where the output will be routed (LST for SYSLST or xxx for the printer associated with logical unit SYSxxx).

- ON - a field to be used for this operation. From 1 to 20 ON fields can be specified.

  - (p,m,f) gives the position, length and format of a numeric or character field. A field must not extend beyond position 32752 or the end of the record.

  - (p,m,f,formatting) gives the position, length and format of a numeric or character field and specifies how the data for this field is to be formatted for printing. A field must not extend beyond position 32752 or the end of the record.

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

    - Mask - an edit mask to be applied to the numeric data. Thirty-three pre-defined edit masks are available, encompassing many of the numeric notations throughout the world with respect to separators, decimal point, decimal places, signs and so on. See Appendix, “Edit Masks for DISPLAY Formatting” on page 43 for complete descriptions and examples of all thirty-three masks. The attributes of each group of masks is shown below.

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

- /x - divide the numeric data by:
  - 1000 (/K)
  - 1000*1000 (/M)
  - 1000*1000*1000 (/G)
  - 1024 (/KB)
  - 1024*1024 (/MB)
  - 1024*1024*1024 (/GB)
- L′string′ - a leading string to appear at the beginning of the character or numeric data column. Each string can be 1 to 10 characters.
- F′string′ - a floating string to appear to the left of numeric data. Each string can be 1 to 10 characters.
- T′string′ - a trailing string to appear at the end of the character or numeric data column. Each string can be 1 to 10 characters.

- (p,m,HEX) gives the position and length of a character field to be printed in hexadecimal format. A field must not extend beyond position 32752 or the end of the record.
- VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record.
- NUM specifies that the record number is to be printed starting at 1 and incrementing by 1 for each record.
- A description of each type of field is given below.

Format Code	Length	Description
BI	1-4 bytes	Unsigned binary
FI	1-4 bytes	Signed fixed-point
PD	1-8 bytes	Signed packed decimal
ZD	1-15 bytes	Signed zoned decimal
CH	1-80 bytes	Character
HEX	1-50 bytes	Character printed as hexadecimal
VLEN	n/a	Record length for VLR (1,2,BI)
NUM	n/a	Relative record number

Optional Operands

- TITLE, PAGE, DATE and TIME - the elements to appear in the title line at the top of each page. Only specified elements appear and in the order given. The title string can be 1 to 50 characters. Alternate formats for date and time can be specified.
- HEADER - a heading to be printed for the corresponding ON field (overriding the ″standard″ heading of ″(p,m,f)″). A heading string can be 1 to 50 characters. NONE can be used to suppress the heading for the corresponding ON field.
- NOHEADER - suppresses the heading line.
- LINES - the number of lines per page (overriding the default of 58).
- BLANK and PLUS - causes the column widths to be dynamically adjusted as needed (overriding the standard fixed column widths) and suppresses leading zeros for numeric fields. BLANK causes a blank, rather than a + to be used for the positive sign.
- TOTAL, MAXIMUM, MINIMUM and AVERAGE - the overall statistics to appear for numeric fields after the columns of data for the report. Only the specified statistics appear and in the order given. Each string can be 1 to 50 characters.
- LIMIT - a limit for the number of invalid decimal values (overriding the default of 200). If n invalid decimal values are found, ICETOOL terminates the operation. See “VERIFY Operator Details” on page 34 for an explanation of invalid decimal values.
- BREAK - the break field to be used to divide the report into sections. Each set of sequential input records with the same break field value is treated as a section in the report. Each section starts on a new page with its own section title and section statistics.
- BTITLE - a string to appear in the section title. The break field and string appear in the section title in the order given. The string can be 1 to 50 characters.
- BTOTAL, BMAXIMUM, BMINIMUM and BAVERAGE - the section statistics to appear for numeric fields after the columns of data for each section. Only the specified statistics appear for each section and in the order given. Each string can be 1 to 50 characters.

Notes

- The control character occupies the first byte of each record.
- Three blanks appear between columns.
- ICETOOL uses CTLCHR=ASA and BLKSIZE=121 to define the file for the printer.

Example of Message Output for DISPLAY

```
      * DISPLAY - prints the values and characters of specific
      * numeric and character fields.  Simple, tailored or
      * sectioned reports can be produced.
      * Example: print a tailored report showing values from the
      * three IN2 file ON fields, with title, heading lines,
      * maximum and minimum, on the printer associated with
      * logical unit SYS015.
        DISPLAY FROM(IN2) LIST(015) BLANK -
        TITLE(′IN2 File Report′) PAGE DATE TIME -
        HEADER(′Store′) ON(50,15,CH) -
        HEADER(′Profit/(Loss)′) ON(26,8,ZD,E1) -
        HEADER(′Employees′) ON(2,3,ZD) -
        MAXIMUM(′Largest′) MINIMUM(′Smallest′)
7T27I DFSORT/VSE CALL 0004 FOR COPY FROM IN2     TO E35 EXIT COMPLETED
7T03I INFORMATION PRINTED ON SYS015
7T28I RECORD COUNT:  000000000000008
7T02I OPERATION RETURN CODE:  00
```

Example of Report Output for DISPLAY

Here′s an example of the output that would appear on the printer associated with SYS015:

```
IN2 File Report        - 1 -        11/29/95        13:15:19

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

Largest             92,378,566          166

Smallest           (64,832,715)          27
```

MODE Operator Details

Syntax

```
MODE STOP
     CONTINUE
     SCAN
```

Function

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

Examples

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

Required Operand

STOP, CONTINUE or SCAN.

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

Note

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

Example of Message Output for MODE

```
      * MODE - specifies the error checking and actions after error
      * detection to be performed for a group of operators.
      * Example: continue processing operators whether or not an
      * error is detected.
         MODE CONTINUE
7T30I MODE IN EFFECT:  CONTINUE
7T02I OPERATION RETURN CODE:  00
```

OCCUR Operator Details

Syntax

```
OCCUR  FROM(filename,...) LIST(LST) ON(p,m,f)   ...
OCCURS                    LIST(xxx) ON(p,m,HEX)
                                    ON(VLEN)
                                    ON(VALCNT)

       TITLE('string') PAGE DATE       TIME
                            DATE(abcd) TIME(abc)

       HEADER('string') ... LINES(n) BLANK ALLDUPS
       HEADER(NONE)                  PLUS  NODUPS
       NOHEADER                            HIGHER(x)
                                           LOWER(y)
                                           EQUAL(v)
```

Function

Prints each unique value for specified numeric and/or character fields and how many times it occurs. From 1 to 10 ON fields can be specified as long as the resulting line length does not exceed 121 bytes. Specifying the BLANK or PLUS operand can compress the columns of data allowing more fields in the report (up to a maximum of 10).

All ON fields specified are used to determine whether a record contains a unique value. A single line is printed for each unique value with the fields printed in columns in the same order in which they are specified.

Simple or tailored reports can be produced. A title line consisting of a string, the page number, the date and/or the time, in any order specified, can appear at the top of each page. Headings can be printed to identify each specified field. Column widths are adjusted automatically according to the operands specified and the length of the fields. Alternate formats for date, time and field values can be specified.

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

Examples

```
OCCUR   FROM(SOURCE,UPDATE) LIST(LST) ON(40,6,CH) ON(VALCNT)
OCCUR   FROM(FAILURES) LIST(015) -
        DATE(YMD.) TITLE('Possible System Intruders') PAGE -
        HEADER(' Userid ') ON(23,8,CH) -
        HEADER(' Logon Failures ')  ON(VALCNT) -
        HIGHER(4) BLANK
```

Required Operands

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.
- LIST - where the output will be routed (LST for SYSLST or xxx for the printer associated with logical unit SYSxxx).
- ON - a field to be used for this operation. From 1 to 10 ON fields can be specified.

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

Format Code	Length	Description
BI	1-4 bytes	Unsigned binary
FI	1-4 bytes	Signed fixed-point
PD	1-8 bytes	Signed packed decimal
ZD	1-15 bytes	Signed zoned decimal
CH	1-80 bytes	Character
HEX	1-50 bytes	Character printed as hexadecimal
VLEN	n/a	Record length for VLR (1,2,BI)
VALCNT	n/a	Value count

Optional Operands

- TITLE, PAGE, DATE and TIME - the elements to appear in the title line at the top of each page. Only specified elements appear and in the order given. The title string can be 1 to 50 characters. Alternate formats for date and time can be specified.
- HEADER - a heading to be printed for the corresponding ON field (overriding the ″standard″ heading of ″(p,m,f)″). A heading string can be 1 to 50 characters. NONE can be used to suppress the heading for the corresponding ON field.
- NOHEADER - suppresses the heading line.
- LINES - the number of lines per page (overriding the default of 58).
- BLANK and PLUS - causes the column widths to be dynamically adjusted as needed (overriding the standard fixed column widths) and suppresses leading zeros for numeric fields. BLANK causes a blank, rather than a + to be used for the positive sign.
- ALLDUPS, NODUPS, HIGHER, LOWER, EQUAL - limits the values to be printed to those whose occurences meet the given criteria. x, y, and v must be specified as n or +n where n can be 1 to 15 decimal digits.

Notes

- The control character occupies the first byte of each record.
- Three blanks appear between columns.
- ICETOOL uses CTLCHR=ASA and BLKSIZE=121 to define the file for the printer.
- JCL statements must be supplied for SORTWK work files to ensure that work space is available for the sort performed for the OCCUR operation.

Example of Message Output for OCCUR

```
      * OCCUR - prints each unique value for specified numeric and
      * character fields and how many times it occurs.
      * Simple or tailored reports can be produced.
      * The values printed can be limited to those for which the
      * value meets specified criteria (e.g. only duplicate values).
      * Example: print a report showing values from the IN1 file
      * ON field, with the number of times each value occurs, on the
      * printer associated with SYS016.
        OCCUR FROM(IN1) LIST(016) TITLE('Dep''t Counts') -
          ON(35,3,CH) ON(VALCNT)
7T27I DFSORT/VSE CALL 0005 FOR SORT FROM IN1     TO E35 EXIT COMPLETED
7T03I INFORMATION PRINTED ON SYS016
7T28I RECORD COUNT:  000000000000072
7T38I NUMBER OF RECORDS RESULTING FROM CRITERIA:  000000000000005
7T02I OPERATION RETURN CODE:  00
```

Example of Report Output for OCCUR

Here's an example of the output that would appear on the printer associated with SYS016:

```
Dep't Counts

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

RANGE Operator Details

Syntax

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

Function

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

Examples

```
RANGE FROM(DATA1) ON(VLEN) HIGHER(52)
RANGE FROM(DATA2,DATA3) ON(25,3,PD) EQUAL(-999)
RANGE FROM(DATA2) ON(25,3,PD) HIGHER(-20) LOWER(+15)
```

Required Operands

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.
- ON - a field to be used for this operation.
  - (p,m,f) gives the position, length and format of a numeric field. A field must not extend beyond position 32752 or the end of the record.
  - VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record.
  - A description of each type of field is given below.

Format Code	Length	Description
BI	1-4 bytes	Unsigned binary
FI	1-4 bytes	Signed fixed-point
PD	1-8 bytes	Signed packed decimal
ZD	1-15 bytes	Signed zoned decimal
VLEN	n/a	Record length for VLR (1,2,BI)

- HIGHER, LOWER, EQUAL, NOTEQUAL - defines the range for the values to be counted. HIGHER and LOWER may be used together or separately. EQUAL and NOTEQUAL must be used separately. x, y, v, and w must be specified as n, +n, or -n where n can be 1 to 15 digits.

Note

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

Example of Message Output for RANGE

```
      * RANGE - prints a message containing the count of values in
      * a specified range for a specified numeric field in one or
      * more input files.
      * Example: print a count of the values in the IN2 file
      * ON field that are higher than -50, but lower than +100.
        RANGE FROM(IN2) ON(20,2,PD) HIGHER(-50) LOWER(100)
7T27I DFSORT/VSE CALL 0006 FOR COPY FROM IN2     TO E35 EXIT COMPLETED
7T28I RECORD COUNT:  000000000000008
7T31I NUMBER OF VALUES IN RANGE FOR (20,2,PD)     :  000000000000004
7T02I OPERATION RETURN CODE:  00
```

SELECT Operator Details

Syntax

```
SELECT FROM(filename,...) TO(filename) ON(p,m,f) ... ALLDUPS
                                       ON(VLEN)      NODUPS
                                                     HIGHER(x)
                                                     LOWER(y)
                                                     EQUAL(v)
                                                     FIRST
                                                     LAST
```

Function

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

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

Examples

```
SELECT  FROM(INPUT) TO(DUPS) ON(11,8,CH) ON(30,44,CH) ALLDUPS
SELECT  FROM(BASE,ADD) TO(OUT1) ON(29,5,ZD) HIGHER(3)
SELECT  FROM(DATAX) TO(NEWEST) ON(135,2,BI) LAST
```

Required Operands

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.
- TO - the file name of the output file. You must supply JCL statements for the output file you specify. You must supply a DEFINE operator if you specify a tape output file or a VSAM output file.
- ON - a field to be used for this operation. From 1 to 10 ON fields can be specified.
  - (p,m,f) gives the position, length and format of a numeric or character field. A field must not extend beyond position 4088 or the end of the record.
  - VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record.
  - A description of each type of field is given below.

Format Code	**Length**	**Description**
BI	1-4 bytes	Unsigned binary
FI	1-4 bytes	Signed fixed-point
PD	1-8 bytes	Signed packed decimal
ZD	1-15 bytes	Signed zoned decimal
CH	1-80 bytes	Character
VLEN	n/a	Record length for VLR (1,2,BI)

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

Note

- JCL statements must be supplied for SORTWK work files to ensure that work space is available for the sort performed for the SELECT operation.

Example of Message Output for SELECT

```
      * SELECT - selects records from one or more input files for
      * inclusion in an output file based on meeting criteria for
      * the number of times specified numeric or character field
      * values occur (e.g. only duplicate values).
      * Example: selects records from the IN1 file, for the SEL1 file,
      * whose ON field occurs only once (i.e, only records with
      * no duplicate ON field values).
        SELECT FROM(IN1) TO(SEL1) ON(30,2,PD) NODUPS
7T27I DFSORT/VSE CALL 0007 FOR SORT FROM IN1     TO SEL1     COMPLETED
7T28I RECORD COUNT:  000000000000072
7T38I NUMBER OF RECORDS RESULTING FROM CRITERIA:  000000000000013
7T02I OPERATION RETURN CODE:  00
```

SORT Operator Details

Syntax

```
SORT FROM(filename,...) TO(filename,...) USE LOCALE(name)
                                             LOCALE(CURRENT)
                                             LOCALE(NONE)
```

Function

Sorts the input files to the output files.

The input files are sorted to the first output file using the DFSORT/VSE control statements in the DFSORT/VSE section. You must supply a DFSORT/VSE SORT control statement to indicate the control fields for the sort. Additional DFSORT/VSE control statements can be used to sort a subset of the input records (INCLUDE or OMIT statement), reformat records for output (OUTREC statement), and so on. See ″DFSORT/VSE Application Programming Guide″ for complete details of DFSORT/VSE control statements.

If the sort is successful, the first output file is copied to the second and subsequent output files (if any).

Examples

```
SORT FROM(IN1,IN2) TO(OUT1,OUT2,OUT3) USE
USTART
  SORT FIELDS=(1,15,CH,A)
  INCLUDE COND=(45,3,CH,NE,C′J69′)
UEND

SORT FROM(VSAMIN) TO(VSAMOUT1) USE LOCALE(FR_CA)
USTART
  SORT FIELDS=(51,4,BI,A,23,8,CH,D)
UEND
```

Required Operands

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.
- TO - the file names of 1 to 10 output files. You must supply JCL statements for the output files you specify. You must supply DEFINE operators for tape output files and VSAM output files you specify.
- USE - indicates that DFSORT/VSE control statements in the DFSORT/VSE section that appears immediately after this SORT statement are to be used for this operation.

Optional Operand

- LOCALE - overrides the installation default for locale processing.

Notes

- JCL statements must be supplied for SORTWK work files to ensure that work space is available for the sort performed for the SORT operation.
- If more than one output file is specified, DFSORT/VSE must be able to read the first output file after it is written in order to copy it to the other output files.
- For maximum efficiency, use a DASD file as the first in a list of output files on both DASD and tape since it is more efficient to write and then read a DASD file than a tape file.

Example of Message Output for SORT

```
      * SORT - sorts one or more input files to one or more
      * output files.
      * Example: sort the IN1 and IN3 files to the OUT1 file using
      * the SORT and OMIT statements in the DFSORT/VSE section.
        SORT FROM(IN1,IN3) TO(OUT1) USE
        USTART
7T61I 01    SORT FIELDS=(35,3,CH,A)
7T61I 02    OMIT COND=(5,6,CH,EQ,C'DELETE')
        UEND
7T27I DFSORT/VSE CALL 0008 FOR SORT FROM IN1     TO OUT1     COMPLETED
7T02I OPERATION RETURN CODE:  00
```

STATS Operator Details

Syntax

```
STATS FROM(filename,...) ON(p,m,f)
                         ON(VLEN)
```

Function

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

Examples

```
STATS FROM(DATA1) ON(VLEN) ON(15,4,ZD)
STATS FROM(VSAMIN) ON(5,4,BI) ON(20,2,PD) -
      ON(12,6,ZD) ON(47,3,FI)
```

Required Operands

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.
- ON - a field to be used for this operation.
  - (p,m,f) gives the position, length and format of a numeric field. A field must not extend beyond position 32752 or the end of the record.
  - VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record.
  - A description of each type of field is given below.

Format Code	Length	Description
BI	1-4 bytes	Unsigned binary
FI	1-4 bytes	Signed fixed-point
PD	1-8 bytes	Signed packed decimal
ZD	1-15 bytes	Signed zoned decimal
VLEN	n/a	Record length for VLR (1,2,BI)

Note

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

Example of Message Output for STATS

```
      * STATS - prints messages containing the minimum, maximum,
      * average, and total for specified numeric fields.
      * Example: print the minimum, maximum, average and total
      * values for the three VLRIN file ON fields.
      * For variable-length records, ON(VLEN) gives statistics about
      * the length of the records.
        STATS FROM(VLRIN) ON(VLEN) ON(12,2,ZD) ON(18,5,PD)
7T27I DFSORT/VSE CALL 0009 FOR COPY FROM VLRIN   TO E35 EXIT COMPLETED
7T28I RECORD COUNT:  000000000000017
7T07I STATISTICS FOR (VLEN)        :
7T08I   MINIMUM:  +000000000000058, MAXIMUM:  +000000000000079
7T09I   AVERAGE:  +000000000000068, TOTAL  :  +000000000001171
7T07I STATISTICS FOR (12,2,ZD)     :
7T08I   MINIMUM:  -000000000000064, MAXIMUM:  +000000000000082
7T09I   AVERAGE:  +000000000000010, TOTAL  :  +000000000000177
7T07I STATISTICS FOR (18,5,PD)     :
7T08I   MINIMUM:  -000000000003892, MAXIMUM:  +000000000018723
7T09I   AVERAGE:  +000000000001127, TOTAL  :  +000000000019168
7T02I OPERATION RETURN CODE:  00
```

UNIQUE Operator Details

Syntax

```
UNIQUE FROM(filename,...) ON(p,m,f)
                          ON(VLEN)
```

Function

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

Examples

```
UNIQUE FROM(INPUT) ON(20,40,CH)
UNIQUE FROM(DATA1,DATA2) ON(5,3,ZD)
```

Required Operands

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.
- ON - a field to be used for this operation.
  - (p,m,f) gives the position, length and format of a numeric field. A field must not extend beyond position 32752 or the end of the record.
  - VLEN is equivalent to ON(1,2,BI) and for variable-length records, represents the record length for each record.
  - A description of each type of field is given below.

Format Code	**Length**	**Description**
BI	1-256 bytes	Unsigned binary
FI	1-256 bytes	Signed fixed-point
PD	1-32 bytes	Signed packed decimal
ZD	1-32 bytes	Signed zoned decimal
CH	1-256 bytes	Character
VLEN	n/a	Record length for VLR (1,2,BI)

Note

- JCL statements must be supplied for SORTWK work files to ensure that work space is available for the sort performed for the UNIQUE operation.

Example of Message Output for UNIQUE

```
      * UNIQUE - prints a message containing the count of unique
      * values for a specified numeric or character field.
      * Example: print the count of unique values in the
      * OUT1 file ON field.
        UNIQUE FROM(OUT1) ON(30,2,PD)
7T27I DFSORT/VSE CALL 0010 FOR SORT FROM OUT1    TO E35 EXIT COMPLETED
7T28I RECORD COUNT:  000000000000040
7T10I NUMBER OF UNIQUE VALUES FOR (30,2,PD)     :  000000000000011
7T02I OPERATION RETURN CODE:  00
```

VERIFY Operator Details

Syntax

```
VERIFY FROM(filename,...) ON(p,m,f) NOSIGN LIMIT(n)
```

Function

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

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

Examples

```
VERIFY FROM(NEW) ON(22,16,PD) ON(7,9,PD)
VERIFY FROM(DATA) ON(28,5,PD) ON(28000,18,ZD) ON(4,7,PD) -
   NOSIGN LIMIT(10)
```

Required Operands

- FROM - the file names of 1 to 9 input files. You must supply JCL statements for the input files you specify. You must supply DEFINE operators for the first input file and for each tape input file you specify.
- ON - a field to be used for this operation.
  - (p,m,f) gives the position, length and format of a numeric field. A field must not extend beyond position 32752 or the end of the record.
  - A description of each type of field is given below.

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

Optional Operands

- NOSIGN - the sign of the decimal values is not to be checked for validity.
- LIMIT - a limit for the number of invalid decimal values (overriding the default of 200). If n invalid decimal values are found, ICETOOL terminates the operation.

Notes

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

Example of Message Output for VERIFY

```
      * VERIFY - examines specified decimal fields in one or more
      * input files and prints a message identifying each invalid
      * value found for each field.
      * Example: identify all values in the two IN2 file
      * decimal ON fields that have invalid digits (A-F)
      * and/or invalid signs (0-9).
        VERIFY FROM(IN2) ON(10,2,ZD) ON(41,6,PD)
7T18A INVALID (10,2,ZD)      VALUE - RECORD:  000000000000003,
      HEX VALUE  FAF2
7T18A INVALID (10,2,ZD)      VALUE - RECORD:  000000000000006,
      HEX VALUE  F134
7T18  INVALID (41,6,PD)      VALUE - RECORD:  000000000000007,
      HEX VALUE  000000105739
7T27I DFSORT/VSE CALL 0011 FOR COPY FROM IN2     TO E35 EXIT COMPLETED
7T28I RECORD COUNT:  000000000000008
7T02I OPERATION RETURN CODE:  12
```

Complex Example

Here′s a sample ICETOOL job that shows how ICETOOL can be used to perform a complex task using multiple operations and files in a single step. This example is completely explained in ″DFSORT/VSE Application Programming Guide″, under ″Examples of DFSORT/VSE Applications″.

```
//  JOB SAMPLE11
//  DLBL IN1,′FLY.INPUT1′,,VSAM,DISP=(OLD,KEEP)
//  DLBL OUTJ69,′%%OUTJ69D′,,VSAM,RECORDS=2000,RECSIZE=63
//  DLBL OUTJ82,′%%OUTJ69D′,,VSAM,RECORDS=2000,RECSIZE=63
//  DLBL DEPTSD,′FLY.OUTPUT1′,,VSAM,RECORDS=2500,RECSIZE=63
//  ASSGN SYS012,181
//  TLBL DEPTST,′FLY.BACKUP1′,,111111,,1
//  DLBL IN2,′FLY.INPUT2′,,SD
//  EXTENT ,339000,,,6500,10
//  DLBL OUT4,′FLY.OUTPUT2′,,SD
//  EXTENT ,339000,,,6600,10
//  DLBL SORTWK1,′SAMP.WORK′,,SD
//  EXTENT ,339000,,,6000,10
//  EXEC ICETOOL,SIZE=100K
* Define input files characteristics
  DEFINE NAME(IN1) TYPE(F) LENGTH(53)
  DEFINE NAME(IN2) TYPE(V) LENGTH(153)
  DEFINE NAME(OUTJ69) TYPE(F) LENGTH(63)
  DEFINE NAME(OUT4) TYPE(V) LENGTH(153)
  DEFINE NAME(DEPTST) UNIT(012)
* Print report showing departments with less than 5 employees
  OCCUR FROM(IN1) LIST(LST) LOWER(5) BLANK -
   TITLE(′Small Departments′) PAGE -
   HEADER(′Department′) HEADER(′Employees′) -
   ON(45,3,CH) ON(VALCNT)
* Copy and reformat selected records
  COPY USE FROM(IN1) TO(OUTJ69)
  USTART
* Select J69 employees and reformat fields
   INCLUDE COND=(45,3,CH,EQ,C′J69′)
   OUTREC FIELDS=(21,10,1X,1,15,45,3,34X)
  UEND
  COPY USE FROM(IN1) TO(OUTJ82)
  USTART
* Select J82 employees and reformat fields and insert text
   INCLUDE COND=(45,3,CH,EQ,C′J82′)
   OUTREC FIELDS=(21,10,1X,1,15,45,3,34X)
  UEND
```

```
* Sort/save the resulting two files
  SORT FROM(OUTJ69,OUTJ82) TO(DEPTSD,DEPTST) USE
  USTART
* Sort by last name, first name
  SORT FIELDS=(12,15,CH,A,1,10,CH,A)
  UEND
* Do following operators even if a previous operator failed,
* but stop processing if a subsequent operator fails.
  MODE STOP
* Verify decimal fields
  VERIFY FROM(IN2) ON(22,6,PD) ON(30,3,ZD)
* Print statistics for record length and numeric fields
  STATS FROM(IN2) ON(VLEN) ON(22,6,PD) ON(30,3,ZD)
* Sort and produce total for each unique key
  SORT FROM(IN2) TO(OUT4) USE
  USTART
* Sort and produce totals in one record for each unique key
   SORT FIELDS=(5,10,CH,A)
   SUM FIELDS=(22,6,PD,30,3,ZD)
  UEND
* Print report containing:
* - key and total for each unique key
* - lowest and highest of the totals
  DISPLAY FROM(OUT4) LIST(LST) -
   TITLE('Unique key totals report') DATE TIME -
   ON(5,10,CH) ON(22,6,PD) ON(30,3,ZD) -
   MINIMUM('Lowest') MAXIMUM('Highest') PLUS
/*
/&
```

Calling ICETOOL from a Program

ICETOOL′s parameter list interface allows you to use information derived by ICETOOL in your program. With this interface, you supply ICETOOL operators and associated DFSORT/VSE statements in a parameter list. ICETOOL prints messages and puts an operation status indicator and ″operation-specific values″ in the ICETOOL parameter list for use by your calling program.

ICETOOL can be called from an assembler program by issuing the LOAD system macro followed by the CALL or ATTACH system macro:

- When ICETOOL is called via the CALL macro, the parameter list pointer must be contained in register 1, and register 2 must contain all zeros. When all operators have been processed, ICETOOL returns to the calling program with the return code field (in the parameter list) and register 15 set to the highest operation return code encountered.
- When ICETOOL is subtasked using the ATTACH macro, the parameter list pointer must be in register 2. When all operators have been processed, ICETOOL returns to the calling program with the return code field (in the parameter list) set to the highest operation return code encountered.

Parameter List Interface

To call ICETOOL using the parameter list interface, you must point register 1 or 2 to a parameter list as follows:

```
                           4 bytes
 +------------+    +-----------------------------+
 | Reg 1 or 2 | -> | X′0000′      | Return Code  |
 +------------+    +-----------------------------+
                   | Statement Area 1 Address    |
                   +-----------------------------+
                   | Return Area 1 Address       |
                   +-----------------------------+
                        .
                        .
                        .
                   +-----------------------------+
                   | Statement Area n Address    |
                   +-----------------------------+
                   | Return Area n Address       |
                   +-----------------------------+
                   | X′FFFFFFFF′                 |
                   +-----------------------------+
```

Use one Statement Area/Return Area pair for each ICETOOL operator you need.

Each Statement Area must consist of:

- A 2-byte length field containing the length of the statement area.
- One ICETOOL operator consisting of one or more 80-character ICETOOL statement images. If the ICETOOL statement contains the USE operand, the DFSORT/VSE section with 80-character DFSORT/VSE control statement images must follow the ICETOOL operator in the area.

Each Return Area must consist of:

- A 2-byte length field containing the length of the return area.

- A 1-byte operation status indicator field to be set by ICETOOL to 0 (operation executed) or 4 (operation not executed).
- One or more 8-byte PD fields to be set by ICETOOL with information specific to the operator specified, as shown on the next page.

Operator	Return Area Length (Bytes)	Operation-Specific Values Returned
COPY	1	None
COUNT	9	Count of records processed
DEFAULTS	1	None
DEFINE	1	None
DISPLAY	9	Count of records processed
MODE	1	None
OCCUR	17	Count of records processed, count of records resulting from criteria
RANGE	17	Count of records processed, count of values in range
SELECT	17	Count of records processed, count of records resulting from criteria
SORT	1	None
STATS	32*n+9	Count of records processed, minimum for ON field 1, maximum for ON field 1, average for ON field 1, total for ON field 1, ... minimum for ON field n, maximum for ON field n, average for ON field n, total for ON field n
UNIQUE	17	Count of records processed, count of unique values
VERIFY	9	Count of records processed

Parameter List Interface Example

Here′s a portion of an assembler program that calls ICETOOL using the Parameter List Interface, and the JCL you might use to run it. Note that the program does further processing based on the information returned by ICETOOL in the parameter list.

```
// JOB EXAMP JOBA,PROGRAMMER
// LIBDEF PHASE,SEARCH=... Sublibrary containing DEPTVIEW
// DLBL IN,'DEPT.BRANCH',,VSAM,DISP=(OLD,KEEP)
// DLBL OUT1,'DEPT.BACKUP1',,VSAM,RECORDS=200,RECSIZE=53
// DLBL OUT2,'DEPT.BACKUP1',,VSAM,RECORDS=200,RECSIZE=53
// EXEC DEPTVIEW,SIZE=128K
DEPTVIEW CSECT
         ...
* SET UP PARAMETER LIST, LOAD AND CALL ICETOOL
         LA    R1,LOADTL        LOAD ICETOOL ENTRY POINT ADDRESS
         LOAD  ICETOOL,(1)      LOAD ICETOOL PROGRAM
         LTR   R15,R15          IF LOAD WAS NOT SUCCESSFUL,
         BNZ   CKLOAD            AN ERROR MESSAGE WILL BE ISSUED
         LR    R15,R1           LOAD ICETOOL ENTRY POINT ADDRESS
         SR    R2,R2            ZERO R2 TO INDICATE CALL INVOKED
         LA    R1,PARLST        LOAD ADDRESS OF PARAMETER LIST
         BALR  R14,R15          CALL ICETOOL
         LTR   R15,R15          IF ANY OPERATIONS WERE NOT SUCCESSFUL,
         BNZ   CKSTAT1            DETERMINE WHICH ONE FAILED
* ALL OPERATIONS WERE SUCCESSFUL
*  CHECK EMPLOYEES PER DEPARTMENT AGAINST ACCEPTABLE LEVEL
         CP    RT4AVG1,EMAVGCK    IF AVERAGE IS ACCEPTABLE,
         BNH   CKQUAL               NO MESSAGE IS NEEDED
* ISSUE A MESSAGE SHOWING AVERAGE, MINIMUM, MAXIMUM, AND
* TOTAL NUMBER OF EMPLOYEES PER DEPARTMENT.
         ...
*  CHECK EXPENSES PER DEPARTMENT AGAINST ACCEPTABLE LEVEL
CKQUAL   CP    RT4AVG2,TLAVGCK    IF AVERAGE IS ACCEPTABLE,
         BNH   PCTCALC              NO MESSAGE IS NEEDED
* ISSUE A MESSAGE SHOWING AVERAGE, MINIMUM, MAXIMUM, AND
* TOTAL EXPENSES PER DEPARTMENT.
         ...
*  CALCULATE THE PERCENTAGE OF DEPARTMENTS OVER/UNDER EMPLOYEE LIMIT
PCTCALC  MVC   WORK+2(4),RT5RCDS+4   COPY NUMBER OF DEPARTMENTS
         SP    WORK+2(4),RT5RNG+4(4) SUBTRACT 'NUMBER WITHIN LIMITS' TO
*                                     GET 'NUMBER OVER/UNDER LIMIT'
         CP    WORK+2(4),P0          IF NONE OVER/UNDER LIMIT,
         BE    PCTPRT                 PERCENTAGE IS ZERO
         MP    WORK+2(4),P100        MULTIPLY NUMBER OVER/UNDER BY 100
         DP    WORK(6),RT5RCDS+4(4)  DIVIDE BY NUMBER OF DEPARTMENTS
* ISSUE A MESSAGE SHOWING THE PERCENTAGE OF DEPARTMENTS THAT ARE
* OVER/UNDER EMPLOYEE LIMIT
PCTPRT   UNPK  PCTVAL,WORK(2)     CONVERT AVERAGE TO PRINTABLE EBCDIC
         OI    PCTVAL+2,X'F0'     ENSURE LAST DIGIT IS PRINTABLE
         ...
```

```
* ONE OR MORE OPERATIONS FAILED
CKSTAT1  CLI   RT1STAT,0          IF OPERATION 1 WORKED,
         BNE   CKSTAT2              CHECK OPERATION 2
* ISSUE MESSAGE: OPERATION 1 FAILED - CHECK ICETOOL OUTPUT
         ...
* LOAD OPERATIONS FAILED
CKLOAD   EQU   *
* ISSUE MESSAGE: LOAD OPERATION FAILED - CHECK ICETOOL OUTPUT
         ...
* PARAMETER LIST
PARLST   DC    A(0)               FLAGS/RETURN CODE
         DC    A(ST1A)            STATEMENT AREA 1 ADDRESS
         DC    A(RT1A)            RETURN AREA 1 ADDRESS
         DC    A(ST2A)            STATEMENT AREA 2 ADDRESS
         DC    A(RT2A)            RETURN AREA 2 ADDRESS
         DC    A(ST3A)            STATEMENT AREA 3 ADDRESS
         DC    A(RT3A)            RETURN AREA 3 ADDRESS
         DC    A(ST4A)            STATEMENT AREA 4 ADDRESS
         DC    A(RT4A)            RETURN AREA 4 ADDRESS
         DC    A(ST5A)            STATEMENT AREA 5 ADDRESS
         DC    A(RT5A)            RETURN AREA 5 ADDRESS
         DC    F′-1′              END OF PARAMETER LIST
* OPERATOR STATEMENT AREAS
*   DEFINE OPERATION
ST1A     DC    AL2(ST1E-ST1)      LENGTH OF STATEMENT AREA 1
ST1      DC    CL80′* DEFINE COPY ATTRIBUTES FOR IN FILE′
         DC    CL80′DEFINE NAME(IN) TYPE(F) LENGTH(53)′
ST1E     EQU   *
*   COPY OPERATION
ST2A     DC    AL2(ST2E-ST2)      LENGTH OF STATEMENT AREA 2
ST2      DC    CL80′* CREATE TWO COPIES OF THE DENVER SITE′
         DC    CL80′* DEPARTMENT RECORDS′
         DC    CL80′COPY FROM(IN) TO(OUT1,OUT2) USE′
         DC    CL80′USTART′
         DC    CL80′* SELECT ONLY THE DENVER SITE DEPARTMENT RECORDS ′
         DC    CL80′ INCLUDE COND=(1,12,CH,EQ,C′′DENVER′′) ′
         DC    CL80′UEND′
ST2E     EQU   *
*   DEFINE OPERATION
ST3A     DC    AL2(ST3E-ST3)      LENGTH OF STATEMENT AREA 3
ST3      DC    CL80′* DEFINE STATS AND RANGE ATTRIBUTES FOR OUT1 FILE′
         DC    CL80′DEFINE NAME(OUT1) TYPE(F) LENGTH(53)′
ST3E     EQU   *
*   STATS OPERATION
ST4A     DC    AL2(ST4E-ST4)      LENGTH OF STATEMENT AREA 4
ST4      DC    CL80′* GET STATISTICS FOR NUMBER OF EMPLOYEES′
         DC    CL80′* AND TRAVEL EXPENSES PER DEPARTMENT′
         DC    CL80′STATS FROM(OUT1) ON(15,2,PD) ON(28,8,ZD)′
ST4E     EQU   *
```

```
*   RANGE OPERATION
ST5A     DC    AL2(ST5E-ST5)      LENGTH OF STATEMENT AREA 5
ST5      DC    CL80′* DETERMINE THE NUMBER OF DEPARTMENTS THAT ARE′
         DC    CL80′* WITHIN THE LIMIT FOR NUMBER OF EMPLOYEES′
         DC    CL80′RANGE FROM(OUT1) ON(15,2,PD) -′
         DC    CL80′ HIGHER(10) LOWER(21)′
ST5E     EQU   *
* RETURN AREAS
*   DEFINE OPERATION
RT1A     DC    AL2(RT1E-RT1STAT)  LENGTH OF RETURN AREA 1
RT1STAT  DS    C                  OPERATION STATUS
RT1E     EQU   *
*   COPY OPERATION
RT2A     DC    AL2(RT2E-RT2STAT)  LENGTH OF RETURN AREA 2
RT2STAT  DS    C                  OPERATION STATUS
RT2E     EQU   *
*   DEFINE OPERATION
RT3A     DC    AL2(RT3E-RT3STAT)  LENGTH OF RETURN AREA 3
RT3STAT  DS    C                  OPERATION STATUS
RT3E     EQU   *
*   STATS OPERATION
RT4A     DC    AL2(RT4E-RT4STAT)  LENGTH OF RETURN AREA 4
RT4STAT  DS    C                  OPERATION STATUS
RT4RCDS  DS    PL8                COUNT OF RECORDS PROCESSED
RT4MIN1  DS    PL8                FIELD 1 - MINIMUM VALUE
RT4MAX1  DS    PL8                FIELD 1 - MAXIMUM VALUE
RT4AVG1  DS    PL8                FIELD 1 - AVERAGE VALUE
RT4TOT1  DS    PL8                FIELD 1 - TOTAL VALUE
RT4MIN2  DS    PL8                FIELD 2 - MINIMUM VALUE
RT4MAX2  DS    PL8                FIELD 2 - MAXIMUM VALUE
RT4AVG2  DS    PL8                FIELD 2 - AVERAGE VALUE
RT4TOT2  DS    PL8                FIELD 2 - TOTAL VALUE
RT4E     EQU   *
*   RANGE OPERATION
RT5A     DC    AL2(RT5E-RT5STAT)  LENGTH OF RETURN AREA 5
RT5STAT  DS    C                  OPERATION STATUS
RT5RCDS  DS    PL8                COUNT OF RECORDS PROCESSED
RT5RNG   DS    PL8                COUNT OF VALUES IN RANGE
RT5E     EQU   *
* VARIABLES/CONSTANTS
WORK     DS    PL6                WORKING VARIABLE
P100     DC    P′100′             CONSTANT 100
P0       DC    P′0′               CONSTANT 0
EMAVGCK  DC    P′17′              ACCEPTABLE AVERAGE EMPLOYEE COUNT
TLAVGCK  DC    P′5000′            ACCEPTABLE AVERAGE TRAVEL EXPENSES
PCTVAL   DS    PL3                PERCENTAGE OF DEPARTMENTS THAT ARE
*                                   OVER/UNDER EMPLOYEE LIMIT
         ...
LOADTL   EQU   *                  ICETOOL ENTRY POINT
         END   DEPTVIEW
/*
/&
```

Appendix. Edit Masks for DISPLAY Formatting

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

- **d** is used to represent a decimal digit (0-9)
- **w** is used to represent a leading sign that will be blank for a positive value or - for a negative value
- **x** is used to represent a trailing sign that will be blank for a positive value or - for a negative value
- **y** is used to represent a leading sign that will be blank for a positive value or (for a negative value
- **z** is used to represent a trailing sign that will be blank for a positive value or) for a negative value

Mask	Pattern	12345678	-1234567
A0	wddddddddddddddd	12345678	-1234567
A1	wddd,ddd,ddd,ddd,ddd	12,345,678	-1,234,567
A2	wddd.ddd.ddd.ddd.ddd	12.345.678	-1.234.567
A3	wddd ddd ddd ddd ddd	12 345 678	-1 234 567
A4	wddd′ddd′ddd′ddd′ddd	12′345′678	-1′234′567
A5	ddd ddd ddd ddd dddx	12 345 678	1 234 567-
B1	wdd,ddd,ddd,ddd,ddd.d	1,234,567.8	-123,456.7
B2	wdd.ddd.ddd.ddd.ddd,d	1.234.567,8	-123.456,7
B3	wdd ddd ddd ddd ddd,d	1 234 567,8	-123 456,7
B4	wdd′ddd′ddd′ddd′ddd.d	1′234′567.8	-123′456.7
B5	wdd′ddd′ddd′ddd′ddd,d	1′234′567,8	-123′456,7
B6	dd ddd ddd ddd ddd,dx	1 234 567,8	123 456,7-
C1	wd,ddd,ddd,ddd,ddd.dd	123,456.78	-12,345.67
C2	wd.ddd.ddd.ddd.ddd,dd	123.456,78	-12.345,67
C3	wd ddd ddd ddd ddd,dd	123 456,78	-12 345,67
C4	wd′ddd′ddd′ddd′ddd.dd	123′456.78	-12′345.67
C5	wd′ddd′ddd′ddd′ddd,dd	123′456,78	-12′345,67
C6	d ddd ddd ddd ddd,ddx	123 456,78	12 345,67-
D1	wddd,ddd,ddd,ddd.ddd	12,345.678	-1,234.567
D2	wddd.ddd.ddd.ddd,ddd	12.345,678	-1.234,567
D3	wddd ddd ddd ddd,ddd	12 345,678	-1 234,567
D4	wddd′ddd′ddd′ddd.ddd	12′345.678	-1′234.567
D5	wddd′ddd′ddd′ddd,ddd	12′345,678	-1′234,567
D6	ddd ddd ddd ddd,dddx	12 345,678	1 234,567-
E1	yddd,ddd,ddd,ddd,dddz	12,345,678	(1,234,567)
E2	yddd.ddd.ddd.ddd.dddz	12.345.678	(1.234.567)

Mask	Pattern	12345678	-1234567
E3	yddd ddd ddd ddd dddz	12 345 678	(1 234 567)
E4	yddd′ddd′ddd′ddd′dddz	12′345′678	(1′234′567)
F1	yd,ddd,ddd,ddd,ddd.ddz	123,456.78	(12,345.67)
F2	yd.ddd.ddd.ddd.ddd,ddz	123.456,78	(12.345,67)
F3	yd ddd ddd ddd ddd,ddz	123 456,78	(12 345,67)
F4	yd′ddd′ddd′ddd′ddd.ddz	123′456.78	(12′345.67)
F5	yd′ddd′ddd′ddd′ddd,ddz	123′456,78	(12′345,67)