TATA CONSULTANCY SERVICES
Experience certainty.

ICETOOL Basics

Contents

- Use of the utility
- Creating Icetool job
- Operators
- Functions performed

Use of the utility

- ICETOOL is a basic DFSORT utility that allows multiple sort operations in a single step

- To understand the full concept of Icetool, we need to be familiar with the operators and operands used and also the way of writing an Icetool job

Creating Icetool job

An ICETOOL job consists of :

- JCL statements which are required for creating any job

- The OPERATOR statements (ICETOOL statements) indicating the operations to be performed

- The additional JCL statements as a result of the specified operator statements

Creating Icetool job contd..

Layout of an ICETOOL job is as below:

```
//PROGA JOB EX01,PROGRAM
//TOOL EXEC PGM=ICETOOL
//TOOLMSG DD SYSOUT=A
//DFSMSG DD SYSOUT=A
//TOOLIN DD *
<ICETOOL statements go here>
/*
<Additional JCL statements go here>
```

Creating Icetool job contd..

JOB - Signifies the beginning of the job.

EXEC - Signifies the beginning of the job step and executes the ICETOOL program with the region of 1024k recommended.

STEPLIB - Defines the library where ICETOOL is present.

TOOLMSG - Defines the output dataset for the ICETOOL messages.

DFSMSG - Defines the output dataset for the DFSORT messages.

TOOLIN - The area where the ICETOOL statements are to be written. The ICETOOL statements in TOOLIN ends with a comment at the end. The additional JCL statements can be prior or after the ICETOOL (or) OPERATOR statements depending on the requirement.

Creating Icetool job contd..

Blank & Comment statements in ICETOOL

Comment statements and blank statements can be placed anywhere among the ICETOOL operator statements.

Comment statements start with an asterisk (*) in column 1.

Blank statements contain blanks in columns 1-72.

```
//TOOLIN DD *
*Commented line
/*
```

Operators

There are 12 main operators used in ICETOOL. They are :

COPY	COUNT
DEFAULTS	DISPLAY
MODE	OCCUR
RANGE	SELECT
SORT	STATS
UNIQUE	VERIFY

Operators contd..

SAMPLE DATASETS

Dataset Name : BRANCH.DATA

City	State	Employees	Revenue	Profit
1 15	**16 17**	**18 21**	**22 27**	**28 33**
Los Angeles	CA	32	22530	−4278
San Francisco	CA	35	42820	3832
Fort Collins	CO	22	12300	−2863
Sacramento	CO	29	42726	8276
Sunnyvale	CA	18	16152	−978

Function performed

The various functions performed by these twelve operators are listed below, the combinations of which can be used to simplify many complex tasks.

COPY : Copies a dataset to one or more output datasets. To create unsorted copies of the input dataset, we can use the COPY operator.

Syntax:

```
//TOOLIN DD *
COPY FROM(ALL) TO(D1,D2,D3)
COPY FROM(ALL) TO(P1) USING(DO1)
/*
//DO1CNTL DD *
INCLUDE COND=(16,2,CH,EQ,C'CA')
```

Operators contd..

where ALL is the DDNAME of the input dataset BRANCH.DATA, D1,D2,D3,P1 are the DDNAMES of the output datasets respectively.

DO1 represents the DDNAME of the control card.

The first COPY operations copies the entire data from BRANCH.DATA to the datasets D1,D2 and D3.

The second COPY operation copies the data from BRANCH.DATA based on some conditions given in the control card.

Only the records which has the data in the field 16 as ‘CA’ is copied to the dataset P1.

Function performed contd..

COUNT : Prints a message containing the count of records in the input dataset. Specifying the keyword COUNT in TOOLIN, counts the records. Though this operator can be used to get the record count alone, generally it is combined with the other operators which also does similar record counting process.

DEFAULTS : Prints the DFSORT installation defaults in a separate list dataset. Specifying keyword DEFAULT in TOOLIN, prints the defaults in the output dataset.

Function performed contd..

DISPLAY : Used to display numeric and character fields from an input data set in readable form. Simple and tailored reports can be generated. Print profit, employees, and city for each Colorado (CO) branch

Syntax:

```
//TOOLIN DD *
DISPLAY FROM(ALL) LIST(OUT) ON(28,6,PD) ON(18,4,ZD) ON(1,15,CH)
//
OUT DD SYSOUT=A
```

From the input dataset BRANCH.DATA, the three fields mentioned above namely PROFIT, EMPLOYEES and CITY are displayed in a readable form in the output dataset with DDNAME OUT. Here the output dataset is the SPOOL.

Function performed contd..

Output displayed in Spool :

(28,6,PD)	(18,4,ZD)	(1,15,CH)
-000000000004278	+000000000000032	Los Angeles
+000000000003832	+000000000000035	San Francisco
-000000000002863	+000000000000022	Fort Collins
+000000000008276	+000000000000029	Sacramento
-000000000000978	+000000000000018	Sunnyvale

Function performed contd..

MODE : Three modes are available which can be set or reset for groups of operators:

STOP mode (the default) stops subsequent operations if an error is detected.

CONTINUE mode continues with subsequent operations if an error is detected.

SCAN mode allows ICETOOL statement checking without actually performing any operations.

Specifying the mode in TOOLIN, does the above mentioned functions.

Function performed contd..

OCCUR : The OCCUR operator shows how many times different field values occur.

There are certain operands used for this. They are :

More than once, that is, duplicate values (ALLDUPS operand)

Only once, that is, non-duplicate values (NODUPS operand)

A specified number of times (EQUAL operand)

More than a specified number of times (HIGHER operand)

Less than a specified number of times (LOWER operand)

Function performed contd..

Syntax:

```
//TOOLIN DD *
OCCUR FROM(ALL) LIST(OUT) -
HEADER('STATE') HEADER('NUMBER COUNT') -
ON(106,4,CH) ON(VALCNT)
/*
```

All the records from the input dataset BRANCH.DATA which has more than one occurrence in the STATE field (16,2) is displayed. VALCNT is a special operator which gives the count of the occurrences.

Here, the output is displayed in spool.

Output :

STATE	**NUMBER COUNT**
CA	3
CO	5

Function performed contd..

RANGE : Range operator is used to count the number of values in the specified numeric field that falls within the range we define. Some of the operands used here are:

EQUAL	Equal to a value
NOTEQUAL	Not equal to a value
LOWER	Less than a value
HIGHER	Greater than a value
HIGHER and LOWER	Greater than a value, but less than another value

Syntax:

```
//TOOLIN DD *
RANGE FROM(ALL) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)
/*
```

Function performed contd..

All the data in the PROFIT field (28,6) which has the value greater than -1500 but less Than 8000 will be displayed in the spool.

Output as in spool :

RANGE FROM(ALL) ON(28,6,PD) HIGHER(-1500) LOWER(+8000)

ICE627I 0 DFSORT CALL 0004 FOR COPY FROM ALL TO E35 EXIT COMPLETED

ICE628I 0 RECORD COUNT: 000000000000005

ICE631I 0 NUMBER OF VALUES IN RANGE FOR (28,6,PD) : 000000000000002

ICE602I 0 OPERATION RETURN CODE: 00

Function performed contd..

SELECT : SELECT operator used to create an output data set with records selected according to how many times different field values occur. The operands used in OCCUR operator as used here as well.

Syntax :

```
//TOOLIN DD *
SELECT FROM(ALL) TO(OUT) ON(18,4,CH) HIGHER(30)
/*
```

All the records from the input dataset BRANCH.DATA which has the data in the Employees field (18,4) with values greater than 30 are selected and given to the output. The output can be sent to spool or collected in a dataset as required. Considering OUT as the DDNAME for the output dataset, there are 2 records with employee count greater than 30, which is routed to the output dataset.

Functions performed contd..

SORT : A single SORT operator can be used to create one to 10 sorted output datasets.

The operands used in SORT operators are :

INCLUDE or OMIT : Used to select or omit subset of input records.

INREC or OUTREC : Used to rearrange the fields of the input records.

OUTFIL : Used to create any number of output datasets with different subset of records and arrangement of fields.

Functions performed contd..

.

Syntax:

```
//TOOLIN DD *
SORT FROM(ALL) TO(OUT,OUT1) USING(DO1)
/*
//DO1CNTL DD *
SORT FIELDS=(16,4,A),FORMAT=CH
INCLUDE COND=(16,2,EQ,C'CO'), FORMAT=CH
/*
```

All the records from the input dataset BRANCH.DATA are sorted based on the State field (16,2) and only the records with state 'CO' are routed to the output datasets. Here there are two DDNAMES OUT, OUT1 which can be spool or dataset, as per requirements.

SORT in ICETOOL has the same behavior as that in DFSORT.

Function performed contd..

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

Syntax :

```
//TOOLIN DD *
STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
/*
```

The statistics of the 3 fields of the records in the input dataset BRANCH.DATA is displayed. This is while working with fixed length records.

'–' used as the continuation character.

Example:

```
STATS -
   FROM(ALL) -
     ON(18,4,ZD) -
       ON(28,6,PD) -
         ON(22,6,PD)
```

Function performed contd..

Output as in Spool:

```
STATS FROM(ALL) ON(18,4,ZD) ON(28,6,PD) ON(22,6,PD)
ICE627I 0 DFSORT CALL 0001 FOR COPY FROM ALL TO E35 EXIT COMPLETED
ICE628I 0 RECORD COUNT: 000000000000005
ICE607I 0 STATISTICS FOR (18,4,ZD) :
ICE608I 0 MINIMUM: +000000000000018, MAXIMUM: +000000000000035
ICE609I 0 AVERAGE: +000000000000025, TOTAL : +000000000000126
ICE607I 0 STATISTICS FOR (28,6,PD) :
ICE608I 0 MINIMUM: -000000000004278, MAXIMUM: +000000000008276
ICE609I 0 AVERAGE: +000000000000798, TOTAL : +000000000003989
ICE607I 0 STATISTICS FOR (22,6,PD) :
ICE608I 0 MINIMUM: +000000000012300, MAXIMUM: +000000000042820
ICE609I 0 AVERAGE: +000000000027306, TOTAL : +000000000136528
ICE602I 0 OPERATION RETURN CODE: 00
ICE601I 0 DFSORT ICETOOL UTILITY RUN ENDED - RETURN CODE: 00
```

Function performed contd..

For variable length records in the dataset:

When working with variable length record data sets, you can use the STATS operator to easily obtain the following information:

- The shortest record in the data set (minimum)
- The longest record in the data set (maximum)
- The average length of records in the data set (average)
- The total number of bytes in the data set (total)

Specify ON(VLEN) in the ON field.

Function performed contd..

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

VERIFY : Examines specified decimal fields in a data set and prints a message identifying each invalid value found for each field.

These operators have no separate syntax. They are used in combination with the operators available for making the operations simpler.

Function performed contd..

SPLICE : Splice is another type if ICETOOL operator which is used for manipulating more than one datasets in the single dataset splicing some data from each part of the input datasets giving out a single output dataset.

Syntax:

```
//TOOLIN DD *
COPY FROM T1 TO INP
COPY FROM T2 TO INP
SPLICE FROM(INP) TO(OUT) ON(FIELD) WITH(OTHER FIELDS) -
USING(DDNAME)
/*
```

T1, T2	- Two different datasets with different data.
INP	- input dataset with different data from several files.
OUT	- output dataset with the spliced data.
FIELD	- the field based on which splicing needs to be done.
OTHER FIELDS	- the remaining fields which needs to appear in the output dataset.
DDNAME	- the control card which has some instructions in it.

The dataset INP now has data from two input datasets.

The field indicated in SPLICE operation specifies that there is some data common between the two sets of data in INP. So, only the record with the common data is written to the output dataset OUT.

The control card gives the data alignment information like INCLUDE, OUREC, OUTFIL NAMES, etc.

SPLICING is one important function of ICETOOL which simplifies even the complex operations.

TATA CONSULTANCY SERVICES
Experience certainty.

TATA

Thank You

IT Services
Business Solutions
Outsourcing