MVT Sort/Merge Control Statements

Control statements for the MVT Sort/Merge facility follow the same conventions as the other MVT utility programs. Column 1 of the control statement must be blank, unless the statement has been given a title, which is usually not the case. Therefore, the text begins in column 2 of the statement. The parameters of the statement must not extend beyond column 71. If the control statement will not fit on a single line, break the statement at a comma and continue the statement on the next line, beginning in any column between columns 2 and 16. A continuation indicator character is <u>not</u> required in column 72 of any statement being continued onto the next line.

Parameters controlling the operation of a SORT or MERGE are submitted on a single control statement. The parameters that may be specified are:

```
FIELDS=(<position>,<length>,<format>,<sequence>,…)
or
FIELDS=(<position>,<length>,<sequence>,…),FORMAT=<format>
```

- <position> is a numeric value specifying the beginning byte of the control field in the input record
- <length> is a numeric value specifying the length in bytes of the control field
- <format> is the format of the data in the control field – valid entries are:
  - ZD for zoned decimal
  - PD for packed decimal
  - BI for binary
  - AC for ASCII character
  - CH for EBCDIC character
  - AQ for EBCDIC character using a specified alternative collating sequence
- <sequence> is either A for ascending or D for descending
- In either format, the parameters inside the parentheses may be repeated as many times as required to specify the composite field that will determine the sort order.
- If all the individual sub-fields that are to be sorted are of the same data format, use of the second syntax will simplify the coding.

```
FILESZ=<x>   (Optional)
```

- <x> is a numeric value specifying an estimated number of records to be sorted

```
SKIPREC=<z>    (Optional; not permitted for MERGE)
```

- <z> is a numeric value specifying the number of
  records to bypass from the input file before beginning
  to select records for sorting

```
EQUALS | NOEQUALS
```

- Specifying EQUALS causes the sort to retain the input
  sequence for records with identical sort control
  fields, whereas NOEQUALS (which is the default)
  indicates that the input order need not be maintained.
  NOEQUALS will result in the best sort performance.

```
CKPT
```

- Checkpoints should be taken

Examples of SORT and MERGE statements:

```
SORT FIELDS=(11,1,CH,A,1,10,CH,A)
```

Sorts on two control fields – 1 byte beginning in byte 11
of the input record and 10 bytes beginning in byte 1 of
the input record; both control fields will be sorted in
ascending sequence and contain EBCDIC character data.

```
SORT FIELDS=(11,1,A,1,10,A),FORMAT=CH
```

This alternative format specifies the same sort as the
first example.

```
SORT FIELDS=(5,4,PD,D,96,20,CH,A),SKIPREC=500,EQUALS
```

Sorts on two control fields – 4 bytes beginning in byte 5
of the input record contains packed decimal data and is
to be sorted in descending sequence and 20 bytes
beginning in byte 96 of the input record contains EBCDIC
character data and is to be sorted in ascending sequence.
The first 500 records read from the input file will be
bypassed (eliminated from the sort and consequently the
output file). Records in which the sort control fields
are identical will be maintained in the order in which
they were read from the input file.

```
MERGE FIELDS=(1,7,CH,A)
```

Merges two or more input files on a single control field – 7 bytes beginning in byte 1 of the input records. The input files must already be ordered in ascending sequence based upon the control field containing EBCDIC character data.

The characteristics of the records in the input file(s) are specified on the RECORD statement:

```
RECORD TYPE=F | V,LENGTH=<l>
```

- TYPE specifies that the input records are either fixed length (F) or variable length (V)
- <l> specifies the input record length for fixed records or the maximum input record length for variable length records.

Alternate Collating Sequence

In order to use an alternating collating sequence, specify AQ as the data format in the SORT or MERGE control fields statement and include an ALTSEQ statement to specify the alternate collating sequence to be used:

```
ALTSEQ CODE=(fftt,fftt,…,fftt)
```

Where ff is the hexadecimal representation of the character whose position is to be changed and tt is the hexadecimal representation of the position to which it is to be moved.

The sort control statements are followed by an END statement to signal that all sort control input has been processed:

```
END
```

MVT Sort/Merge Parameters

Parameters controlling the overall performance of the Sort/Merge program are submitted using the PARM= parameter on the execution Job Control Card. The parameters that may be specified are:

```
BALN
```

- Utilizes the Balance Tape or Direct Access Technique for sort work distribution.

```
CORE=<yyyyy>
```

- Main Storage Allocation, where <yyyyy> specifies the number of bytes to be allocated for the Sort/Merge program including buffers.

```
CRCX
```

- Utilizes the Criss Cross Direct Access Technique for sort work distribution.

```
DIAG
```

- Creates detailed diagnostic messages for use in solving severe Sort/Merge problems.

```
MSG=AC | AP | CC | CP | NO
```

- Sort message output, where:

    AC specifies all messages to the console
    AP specifies all messages to SYSLOG
    CC specifies critical messages to the console
    CP specifies critical messages to SYSLOG
    NO specifies no messages are to be output

  The default is NO, which will result in no messages. If you receive a non-zero return code from the sort, you should resubmit the sort with a PARM='MSG=AP' so that diagnostic messages will be produced.

```
OSCL
```

- Utilizes the Oscillating Tape Technique for sort work distribution.

```
POLY
```

- Utilizes the Polyphase Tape Technique for sort work distribution.