Utilities: SyncSort

EI&O Document ID: D0073
Last Updated: 06/27/2002

This document briefly describes how to use SyncSort in z/OS (OS/390) to sort an OS data set. It describes the SYCSORT cataloged procedure, lists required ddnames used by SyncSort, gives an example of sorting a file, and includes how to obtain SyncSort manuals.

Related search terms include utilities, JCL, PL/I, Assembler, and COBOL.

UF Information Technology

UFIT

2046 NE Waldo Rd, Suite 2100
Gainesville Florida 32609-8942
(352) 392.2061
<editor@cns.ufl.edu>
# Table of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SyncSort</td>
<td>3</td>
</tr>
<tr>
<td>DD Names</td>
<td>3</td>
</tr>
<tr>
<td>REGION Size</td>
<td>4</td>
</tr>
<tr>
<td>The SYNCSORT Cataloged Procedure</td>
<td>4</td>
</tr>
<tr>
<td>Basic JCL for SyncSort</td>
<td>4</td>
</tr>
<tr>
<td>Program-Initiated Sorts</td>
<td>5</td>
</tr>
<tr>
<td>How to Specify Options</td>
<td>5</td>
</tr>
<tr>
<td>SyncSort Control Statements</td>
<td>6</td>
</tr>
<tr>
<td>Sorting Example</td>
<td>7</td>
</tr>
<tr>
<td>Documentation</td>
<td>8</td>
</tr>
<tr>
<td>Questions</td>
<td>8</td>
</tr>
</tbody>
</table>
SyncSort

SyncSort is the only general-purpose sort program supported under z/OS (OS/390) at EI&O. You can use SyncSort any of the following ways:

- through the SYNCSORT cataloged procedure,
- by invoking it in your own JCL (// EXEC PGM=SORT,...),
- from a program written in Assembler, COBOL, or PL/I.

This document primarily describes how to use the SYNCSORT cataloged procedure. You should have a general familiarity with JCL and/or one of the languages mentioned above.

SyncSort has the following functions:

- sort -- to rearrange data set records in chronological or alphabetical order, producing a specific sequential form
- merge -- to combine as many as 32 sequential data sets into 1 data set under the same sequence
- copy -- to reproduce a file, bypassing the sorting process entirely

DD Names

SyncSort uses the following ddnames:

- **SORTIN** defines the input data set for a sorting application; not required for a merge-only operation.
- **SORTIN01-- SORTIN16** define the input data sets for a merging application; not required for a sorting application.
- **SORTWK01-- SORTWK16** define intermediate storage data sets for a sorting application; not required for a merge.
- **SORTOUT** defines the output data set for a sorting or merging application; should not be the same as the SORTIN data set because a system failure could then destroy both your output and your data.
- **SYSOUT** is used as the system message data set.
- **SYSIN** defines the application control data set.
REGION Size

SyncSort uses the default system region size unless you specify a region size. The current system default at EI&O is 1M (1024K). SyncSort allows you to specify up to 4M by using the REGION= parameter on the JOB or EXEC statement.

The SYNCSORT Cataloged Procedure

The SYNCSORT procedure is the only supported sort procedure. Figure 1 shows the JCL for SYNCSORT.

You need to supply the SORTIN, SORTOUT, and SYSIN DD statements.

Figure 1. JCL for the SYNCSORT Procedure.

```
//jobname JOB (,,time,lines),'your name',CLASS=class
/*ROUTE PRINT node.location
//SORT EXEC SYNCSORT
//SORTIN DD ...input data set...
//SORTOUT DD ...output data set...
//SYSIN DD *
... sort control statements ...
```

The following optional keyword parameters can be added to the EXEC statement:

- **OPTION='options'**
  
  The default is no options. Add this parameter to the EXEC statement as in shown below to specify any desired SyncSort options. For valid options, see the "Options" section below.
  
  ```
  //SORT EXEC SYNCSORT,OPTION='options'
  ```

- **REGION=nM**
  
  Use this keyword to specify the region size for the SORT utility. For example,
  
  ```
  //SORT EXEC SYNCSORT,REGION=2M
  ```

- **WKVOL=vol**
  
  By default, the sortwork data sets will be allocated on the work packs. Use this parameter to specify a particular volume.

Basic JCL for SyncSort

Figure 2 shows the basic JCL needed to sort a data set if you do not use the SYNCSORT cataloged procedure.

If you have a very large data set, you may need to request more than 5 cylinders of work space. Also, do NOT specify more SORTWKnn DD statements than there are temporary packs available. There are currently four. This causes negative system and job performance and may result in your job being cancelled.
Figure 2. JCL for Using SyncSort.

```plaintext
//SORTIT JOB (,,time,lines),'your name',CLASS=class
//ROUTE PRINT node.location
//SORT EXEC PGM=SORT,PARM='options'
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(5,3))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(5,3))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(5,3))
//SORTIN DD DSN=your.input.data.set,DISP=OLD
//SORTOUT DD ... DD statement for output data set ...
//SYSIN DD *
... SyncSort control statements
```

Program-Initiated Sorts

Refer to the documentation for Assembler, COBOL, or PL/I for information on how to invoke SyncSort from within a program.

How to Specify Options

Using the `SYNCSORT` cataloged procedure, you can add an OPTION parameter to the EXEC statement as follows:

```plaintext
// EXEC SYNCSORT,OPTION='options'
```

If you are using your own JCL rather than using the `SYNCSORT` cataloged procedure, you can specify options in the PARM parameter of the EXEC statement, as in this example:

```plaintext
// EXEC PGM=SORT,REGION=region,PARM='options'
```

For program-initiated sorts, use the `$ORTPARM DD` statement, as in this example:

```plaintext
//$ORTPARM DD *
CORE=MAX-32K X
RESET
```

Parameters and Options

The following are SyncSort default parameters, which you can change as described above.

- **BMSG**
  - BMSG lists alternate parameters specified through `$ORTPARM` on the output message data set.

- **CMP=CLC**
  - This default causes limited validation of the data contained in
fields designated as packed or zoned decimal. To have the sort terminate if data in PD or ZD fields are invalid, specify CMP=CPD.

**CORE=MAX-44K**

For program-invoked sorts with exit routines, you must determine the amount of memory needed by the exits, then specify CORE=MAX-nnnK.

**NOSNAP**

NOSNAP will suppress the SyncSort SNAP dump, which is produced in the event of a critical error. If you have a SYSUDUMP DD statement, you will receive the normal OS dump. If you require a SNAP dump, specify the DEBUG parameter.

**NOEQUALS**

For a more efficient sort, the order of the records is not preserved. If the order needs to be retained, use the EQUALS keyword.

**NOIOERR**

Error messages are issued when an I/O error occurs. If IOERR=ABE is specified, a USER 999 ABEND will occur.

**LIST**

Control statements will be listed on the SYSOUT data set. NOLIST will turn this off.

**MSGDD=SYSOUT**

The default DD name of the message data set is SYSOUT. Any other valid DD name can be substituted.

**PRINT121**

To be compatible with the display options of COBOL exit routines, the defined characteristics of the message data set are: DCB=(LRECL=121,BLKSIZE=121,RECFM=FA)

**NORC16**

An unsuccessful sort will issue a return code of 16. If RC16=ABE is specified, a USER 16 ABEND will occur.

**RELEASE=ON**

Unused space in sortwork data sets will be released during execution time unless the sort is program-initiated.

**NORLSOUT**

To release excess space on the SORTOUT data set, RLSOUT can be specified.

**no RESET**

RESET must be given as a parameter if the SORTOUT data set is a VSAM file that was created with the REUSE option to prevent SyncSort from treating SORTOUT as a MOD data set.

**SECOND=ON**

A secondary allocation is given to the sortwork data sets if needed.

**VLTEST=0**

No validity testing of variable-length records will be done.

**SyncSort Control Statements**

SyncSort uses the following program control statements: SORT, MERGE, ALTSEQ, END.
For a sort application, the `SORT` statement specifies the sort order. The general form of the `SORT` statement is as follows:

```
SORT FIELDS=(sort key, sort key, ..., sort key), FILSZ=E
```

in which "n" is an estimate of the number of records to be sorted. Sort keys specify the fields within the record upon which to sort, their data types, and whether they are to be sorted in ascending or descending order.

Each sort key has the following four parts: starting byte, length in bytes, format, and order.

- **start** specifies the starting byte position in the record relative to byte 1, which is the first byte. For variable-length records, the offset must include the record descriptor, so the first byte of the data portion of the record is position 5.

- **length** specifies the length of the field in bytes.

- **format** must be a valid data type, such as one of the following:
  - CH: EBCDIC characters
  - ZD: zoned decimal
  - PD: packed decimal
  - FI: signed integers
  - FL: floating point
  - AC: ASCII characters
  - BI: unsigned binary
  - AQ: alternate sequence as specified by the ALTSEQ control statement

- **order** is either A for ascending order, D for descending order, or E as modified by an E6i exit.

## Sorting Example

The following example helps to illustrate how to `SORT` a data set. Suppose you have a card-image (80-byte records) data set called UF.userid.GRADES, which is to be sorted alphabetically by student name within the same letter grade. Assume the name is in columns 1-25 and the letter grade is in column 50 of each record, and there are approximately 100 students in the file. Figure 3 shows a sample job setup using the `SYNCSORT` cataloged procedure.
The input data set is UF.userid.GRADES. The output data set will be sent to the printer.

Figure 3. Example to **SORT** a Data Set.

```plaintext
//SORTLST JOB (,,time,lines),'your name',CLASS=class
/*ROUTE PRINT node.location
//SORT EXEC SYNSORT
//SORTIN DD DSN=UF.userid.GRADES,DISP=SHR
//SORTOUT DD SYSOUT=A
//SYSIN DD *
SORT FIELDS=(50,1,CH,A,1,25,CH,A),FILSZ=E100
```

**Documentation**

Documentation for the SyncSort utility is in the SyncSort Release 3.4 Programmer's Guide from SyncSort, Inc. You can order manuals from the vendor at this address:
SyncSort Incorporated
50 Tice Boulevard
Woodcliff Lake, NJ 07675
Phone 201/930-9700

e-mail: mvs_tech@syncsort.com
Fax: (201) 930-8284
Attn: Syncsort Product Services

**Questions**

If you need assistance using SyncSort at EI&O, contact the EI&O Support Desk by phone at (352) 392-2061 or e-mail to consult@lists.ufl.edu.

**Your Comments are Welcome**

We welcome your comments and suggestions on this and all UFIT documentation. Please send your comments to:

**UF Information Technology**

**UFIT**

2046 NE Waldo Rd, Suite 2100
Gainesville Florida 32609-8942
(352) 392.2061
<editor@cns.ufl.edu>