

Unicenter[®] CA-Sort[®] for VSE

User Guide

9.0



Computer Associates™

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

This documentation may not be copied, transferred, reproduced, disclosed or duplicated, in whole or in part, without the prior written consent of CA. This documentation is proprietary information of CA and protected by the copyright laws of the United States and international treaties.

Notwithstanding the foregoing, licensed users may print a reasonable number of copies of this documentation for their own internal use, provided that all CA copyright notices and legends are affixed to each reproduced copy. Only authorized employees, consultants, or agents of the user who are bound by the confidentiality provisions of the license for the software are permitted to have access to such copies.

This right to print copies is limited to the period during which the license for the product remains in full force and effect. Should the license terminate for any reason, it shall be the user's responsibility to return to CA the reproduced copies or to certify to CA that same have been destroyed.

To the extent permitted by applicable law, CA provides this documentation "as is" without warranty of any kind, including without limitation, any implied warranties of merchantability, fitness for a particular purpose or noninfringement. In no event will CA be liable to the end user or any third party for any loss or damage, direct or indirect, from the use of this documentation, including without limitation, lost profits, business interruption, goodwill, or lost data, even if CA is expressly advised of such loss or damage.

The use of any product referenced in this documentation and this documentation is governed by the end user's applicable license agreement.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227-7013(c)(1)(ii) or applicable successor provisions.

© 2002 Computer Associates International, Inc. (CA)

All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contents

About This Guide	v
Chapter 1. Introduction	1-1
1.1 Features and Functions	1-2
1.2 Program Organization	1-3
1.3 Supported Devices	1-5
1.4 Sort/Merge Specifications	1-6
Chapter 2. Using the Control Language	2-1
2.1 Overview	2-1
2.2 Control Statement Format	2-2
2.3 Continuation Records	2-5
2.4 Control Statement Types	2-7
2.5 Sort/Merge Control Statements	2-11
Chapter 3. Defining User Exits	3-1
3.1 Overview	3-1
3.2 Loading User Exit Routines	3-5
3.3 Parameter Conventions for User Exit Routines	3-8
3.4 User Exit Parameters and Functions	3-9
3.5 Phase 1 Exits	3-12
3.6 Phase 2 Exits	3-18
3.7 Phase 3 Exits	3-20
3.8 Examples of User Exits	3-29
Chapter 4. Operating the Sort/Merge	4-1
4.1 Overview	4-1
4.2 Initiation by Job Control Statements	4-1
4.3 Year 2000 New Keyword Examples	4-22
4.4 Initiation from an Executing Program	4-43
4.5 Unicenter CA-Sort Extended Features	4-51
Glossary	X-1
Index	X-3

About This Guide

This guide serves as the primary reference for the general user of Unicenter CA-Sort Release 9.0. It explains how to use Unicenter CA-Sort and contains examples for illustration.

Organization

Chapter	Description
1	Introduces Unicenter CA-Sort features, program organization, system requirements and SORT/MERGE specifications.
2	Instructs the user in the SORT/MERGE Control Language. Each Control Statement is discussed in detail.
3	Contains all information on SORT/MERGE User Exits. Each Exit is discussed in detail.
4	Describes the methods and procedures used to initiate execution of Unicenter CA-Sort.
Glossary	Defines terms used in the manual.
Index	Provides a quick way to locate specific material.

Summary of Revisions

Unicenter CA-Sort 9.0 allows the use of data spaces as work files. Users may now elect to use data spaces instead of work files. This feature can be implemented using the new DSPSIZE parameter available on the CASORT macro and on the Unicenter CA-Sort OPTION control statement.

Unicenter CA-Sort and Unicenter CA-SRAM Publications

In addition to this guide, the following publications are supplied with Unicenter CA-Sort:

Name	Contents
Unicenter CA-Sort Installation Guide	Installation procedure to install Unicenter CA-Sort
Unicenter CA-Sort Systems Programmer Guide	Planning, installation, and maintenance procedures.
Unicenter CA-Sort/Unicenter CA-SRAM Message Guide	Error messages and return codes. For each message an appropriate reason and possible actions are listed.
Unicenter CA-SRAM User Guide	Complete information on all of Unicenter CA-SRAM available to the general user, including command syntax.
Unicenter CA-SRAM Installation Guide	Installation procedure to install Unicenter CA-SRAM.
Unicenter CA-SRAM Systems Programmer Guide	Planning, installation, and maintenance procedures.

All guides are updated as required. Instructions accompany each update package. Technical changes are identified by a revision bar (|) in the left margin. Revision bars are not used for editorial changes and new manuals.

Related Publications

The following publications relate to Unicenter CA-Sort and are available from Computer Associates:

- CA-CIS documentation set

The following publications are not produced by Computer Associates but are referenced in this publication or recommended reading:

- IBM's Using VSE/VSAM Commands And Macros
- IBM's VSE/ESA Data Management Concepts
- IBM's VSE/ESA Macro Reference
- IBM's VSE/ESA Macro Users Guide
- IBM's VSE/ESA System Management Guide
- IBM's VSE/ESA System Control Statements
- IBM's VSE/VSAM Programmer's Reference
- IBM's Using VSE/VSAM Space Management For SAM

Command Notation

Enter the following exactly as they appear in command descriptions:

UPPERCASE	Identifies commands, keywords, keyword values and operators which must be coded exactly as shown.
MIXed Cases	Identifies command abbreviations. The uppercase letters are the minimum abbreviation; lowercase letters are optional.
Delimiters	, () = and blanks

The following clarify command syntax; do not type these as they appear:

lowercase	Indicates keyword values which you must supply.
[]	Identifies optional keywords.
{ }	Requires choosing one of the keywords specified.
	Separates alternative keywords or values. Choose one.
...	Means the preceding value may be repeated more than once.

Sample Command	Explanation
Sort [WORK ={ n DA }]	Sort is operator keyword. WORK is optional operand. You must either give a value for n or specify DA .

Chapter 1. Introduction

Computer Associates has been an industry leader in the study and development of sorting methods. Throughout the continuing evolution of Unicenter CA-Sort, new techniques have been analyzed and sorting performance evaluated using simulation and actual field test results. Based on this research, numerous improvements and optimizations have been applied to the existing sorting techniques. For example, by upgrading sorting to the level of an access method, the Unicenter CA-Sort Sort/Merge system is capable of performing multiple concurrent sorts. Thus, the historical restriction that only one sort can be performed at a time has been eliminated.

The conventional sorting methods used by the majority of today's computing systems have been improved as well, by maximizing performance and minimizing system resource usage, especially in multiprogramming environments.

The Unicenter CA-Sort system has been designed to support any IBM VSE installation. It consists of two major parts:

- Unicenter CA-Sort Sort/Merge
- Unicenter CA-SRAM (Sort Reentrant Access Method).

Unicenter CA-Sort Sort/Merge is a system which runs as a normal utility program and may be initiated using standard job control statements or from user application programs. It is functionally compatible with the IBM packages 5746-SM1 (DOS/VS) and 5746-SM2 (DOS/VS). The user may access some important logical parts of the system by special exits. This software normally handles all peripheral I/O device operations.

Unicenter CA-SRAM is a powerful programming tool that is suitable for internal program sorting in various programming languages. The system may be re-entered several times within the same program execution. This differs from the purely sequential operation of the conventional sort/merge product. The sequence of sort calls for the reentrant system is comparable to an access method, reading and writing sorted data to and from its work file. The sorted records remain in the work file as sorted sequences after processing, and may be retrieved again in the same or another program without rereading and resorting the data. Thus, a considerable amount of I/O time and external output storage is saved. For more information, see the accompanying Unicenter CA-SRAM guides.

1.1 Features and Functions

The Unicenter CA-Sort program:

- Sorts up to nine files into 1-9 output files, each with a user-specified collating sequence.
- Merges up to nine previously sequenced files into one sequential output file.
- Copies files with possible selection and reformatting of logical records.
- Provides exits to allow for linkage to user routines. At these exits, the user can perform initialization or termination processing, record modification, or I/O error processing.
- Requires work files only when the input file is too large to be sorted entirely in-core.
- May be used as a sophisticated test data generator or auditing tool by allowing selection of records to be sorted from any portion of the input file.
- Functions on any VSE/SP 2 system and above.
- Supports year 2000 features, including the Y2PAST installation and run-time option, which allows 2-digit character, zoned decimal, packed decimal or decimal year data to be handled correctly by establishing a fixed or sliding century window.

Additional benefits may be gained when used with Unicenter CA-Dynam products, including:

- Dynamic SORTWK file allocation.
- Dynamic SORTWK/SORTOUT file secondary allocation.
- Automatic SORTWK/SORTOUT file truncation.
- SORTIN file LRECL, BLKSIZE and RECFM specification are not required.
- Cataloged symbolic SORT control fields.

1.2 Program Organization

The Unicenter CA-Sort system is fully relocatable, may be executed in any VSE partition, and optimizes virtual storage management.

Initiation of Unicenter CA-Sort is under the user's control. The system may be started either by the EXEC job control statement in the application JCL, or by invoking Unicenter CA-Sort in any programming language.

The Unicenter CA-Sort sort/merge program consists of 4 parts (phases). Each nonresident SVA phase is loaded from the core image library and adapted internally according to the sort/merge techniques defined by the user.

Phase 0 (Initialization and Optimization Phase)

Phase 0 will be executed in every SORT and MERGE execution. During this phase, the JCL and control statements submitted (or the parameter list for invoked sorts or merges) will be checked for errors in syntax or conflicts with system information. Error messages will be printed on the printer and console if detected. In addition, the sorting environment is checked carefully. Internally, execution variables and capacity estimates are made, based upon the CPU, storage available, disk devices on the system, and other similar factors. In general, Unicenter CA-Sort optimizes itself to run most efficiently using information gathered in this phase.

Phase 1 (Input and Internal Sorting Phase)

Phase 1 will be executed in every SORT execution. During this phase, the input files are opened and records are read and placed into sorted sequences called 'strings' in memory. A new string is begun when a record will not logically follow the previous record in sequence. Whenever possible, Unicenter CA-Sort attempts to keep these strings entirely in main memory, thus avoiding the use of intermediate work files.

If there is not enough storage in the partition to hold all the strings, the work file is then opened and the sorted sequences written to it. The creation of sorted strings from the input data is the primary function of Phase 1.

Phase 2 (Intermediate Merge Phase)

Phase 2 will be performed only for SORT executions where the number of strings created in Phase 1 is greater than the 'merge order.' Merge order is the maximum number of strings that may be combined at a time and is related to the amount of memory available. The main purpose of Phase 2 is to merge short strings into longer ones, thus reducing the total number of strings. Phase 2 will end when the number of strings remaining is less than the merge order and can be merged into a single string during the output phase. Small sorts normally skip this phase entirely.

Phase 3 (Final Merge and Output Phase)

This phase is executed for every SORT and MERGE task. During Phase 3, the output file is opened and the strings in memory or from the work files are combined in sequence to form a single sorted string that is written to the output device.

1.3 Supported Devices

As shown in the table below, input and output files may reside on any combination of magnetic tape and CKD or FBA direct access storage devices. Input devices may be mixed in any combination for a sort, but no more than two device types may be mixed as input to a merge-only execution at a time.

Intermediate work files may reside on any combination of CKD and FBA disk devices supported on your system. Tape devices are not supported for intermediate storage.

Table 1-1. Supported Devices

File	Type	Model
Input, Output, Work	CKD DISK	2311
		2314/2319
		3330
		3340/3344
		3350
		3375
		3380
		3390
		9345
Input, Output, Work	FBA DISK	3310
		3370
		93xx Series
Input, Output only	TAPE	2400 Series
		3400 Series
		3480 Cartridge
		3490 Cartridge
		8809/ S/S

Rotational Positional Sensing (RPS) will be used on any disk device that supports it if RPS=YES is generated in the supervisor.

1.4 Sort/Merge Specifications

1.4.1 Minimum System Requirements

Unicenter CA-Sort requires the following minimum hardware configuration:

- 30K bytes of main storage, excluding the VSE system routines and any user exit routines, to operate a merge or the string-merge sort algorithm. Depending on the input and output file block size and the logical record length, additional storage will be required.
- standard instruction set (370, 4300, or ESA).
- if disk input/output or intermediate storage is required:
one 2311, 2314, 2319, 3330, 3330-11, 3340/3344, 3350, 3375, 3380, 3390, 3310, 3370, or 93xx disk unit.
- unit record equipment for:
SYSLST, SYSIPT and SYSLOG which may be assigned to any device supported by IBM device-independent I/O (DIMOD).
- if tape input or output is desired:
one 2400, 3400, 3480, 3490 or 8809 (start/stop mode) series tape unit.

Unicenter CA-Sort performance improves when more than the minimum machine configuration is used. The maximum hardware configuration for Unicenter CA-Sort is as follows:

- 8Mb of main storage, plus the size of the user's program.
- nine CKD or FBA disk units in any combination for primary extents of intermediate storage. The secondary allocations may be spread across any number of volumes. No more than 255 extents may be allocated on all work files.
- nine devices for input to a sort or merge. Each tape unit may have any number of alternates, up to the VSE system limit.

1.4.2 SORT Specifications

The general specifications for a SORT are:

- Up to 32 control fields with different data formats and sort sequences may be specified. The sum of the control fields may not exceed 4092 bytes.
- A maximum of nine input files may be sorted, and may be any combination of devices, organizations and blocking factors. Tape input files may consist of any number of volumes.
- A maximum of nine output files sorted in different collating sequences may be produced by one sort invocation. When multiple output files are created, all must be of the same generic device class (all tape, all CKD disks, all FBA disks, or all VSAM).
- Minimum and maximum physical record lengths are shown in the second table. If the amount of main storage assigned to the program decreases, the maximum block size may also decrease.
- Intermediate storage may consist of a maximum of 255 extents allocated among one to nine work files. The intermediate storage may be allocated on any mixture of CKD and FBA devices supported on your system. VSAM-managed SAM files under VSE may be used for intermediate storage.
- When input is from direct access devices, output may optionally be record addresses (Tag Sort). This output may also include control fields.
- When records with all control fields equal are sorted, the output order of these records is unpredictable, unless the EQUALS parameter is specified (see the "SORT Control Statement" chapter).
- Unicenter CA-Sort will sort null files. Note that a 'null' file is defined as a data set having a format 1 label in the VTOC and whose first data record has a data length of zero.

1.4.3 MERGE Specifications

The following are general specifications for a MERGE:

- Input to a merge may be fixed-length, variable-length, variable-length spanned, blocked or unblocked. All records for a given merge must have the same formats. Different blocking factors are allowed.
- Minimum and maximum physical record length restrictions may be found in the second table.
- Up to nine input files may be merged.
- No more than two generic device classes (CKD disk, FBA disk, tape, VSAM) may be mixed as input to a merge-only operation.
- The organization of the input and output files may be sequential, VSAM or ISAM (input only).
- The merge program has exits to allow user routine linkage. At these exits, user routines can perform initialization or termination processing, record modification, record substitution or I/O error processing.
- A special COPY function permits the copying of sequential, VSAM and direct access files. ISAM data sets may not be copied by means of the COPY function.

1.4.4 File Formats and Organizations

All VSE logical file formats are accepted as follows:

- Fixed-length record files consisting of logical records which may be blocked. The blocks within each file may contain varying numbers of logical records; however, short blocks will be accepted only if they are a multiple of the record length.
- Variable-length record files consisting of physical blocks with varying size. The record and block length information is contained in the first four bytes of each record and block.
- Variable-length spanned record files containing blocks of varying size (preferably fixed-length blocks), where any logical record may span one or more blocks. Unicenter CA-Sort accepts these records from any device on input, but it can use only DASD devices for output of spanned records. Tape output files will always contain nonspanned records.

Input and output file organizations accepted by Unicenter CA-Sort are as follows:

- Sequential (SAM) input and/or output
- ISAM input only
- VSAM input and/or output (VSAM files may have KSDS, ESDS or RRDS organizations)
- VSAM-managed SAM input and/or output.

Table 1-2. Maximum Logical Record Lengths

SORTIN Device Type	SORTWK Device Type	Sort Minimum	Sort Maximum	Merge Minimum	Merge Maximum
Tape	CKD	12	SORTWK track capacity - 8	12	32,767
	FBA	12	32,760		
CKD	CKD	1	If SORTWK track capacity > SORTIN track capacity, limit is SORTIN track capacity. Else, SORTWK track capacity - 8	1	SORTIN track capacity
	FBA	1	32,760		
FBA	CKD	1	If SORTWK track capacity < 32,760, limit is SORTWK track capacity - 8. Else, 32,760.	1	32,761
	FBA	1	32,760		
VSAM	CKD	1	If SORTWK track capacity < 32,760, limit is SORTWK track capacity - 8. Else, 32,760	1	32,767
	FBA	1	32,760		

1.4.5 Notes on Maximum Block Size and Record Length

- The second table contains theoretical values which may be restricted due to lack of main storage.
- If the user expands the internal record length in an exit (as specified by the l^2 parameter in the RECORD statement), the record length for fixed-length records must not be extended beyond the values given in the 'logical record length' positions in the second table. Variable-length records may be extended up to 32,764 bytes, independently of the work device used if spanned records are generated.
- Large block sizes increase the amount of main storage the sort or merge will need, and tend to decrease the amount of storage for the actual sort or merge algorithm. This applies also to large CISIZES with FBA devices, and may adversely affect performance.
- Because the optimum value for work file block size is chosen based upon the smallest capacity device in the DISK= parameter of the CASORT installation macro, the limiting factor for input record length is the track size of the lowest capacity device in the macro. This restriction applies in all cases, even when a sort will be performed entirely in-core.

The second table may be summarized as follows:

- If the output file is to a CKD device, the block size cannot exceed the track capacity of the device chosen.
- The record length being sorted cannot exceed the track capacity of the smallest capacity workfile device.
- Maximum block size is limited to the track size if the input file resides on a CKD device, and limited to 32,760 if the input file resides on an FBA device.

Chapter 2. Using the Control Language

2.1 Overview

Unicenter CA-Sort requires control statements which describe:

- the function to be performed
- control field definitions
- modifications to be made by user-written exit routines
- the input and output data file parameters.

The statements are checked for validity. If the program finds an error, it issues a diagnostic message. Refer to the *Unicenter CA-Sort and Unicenter CA-SRAM Message Guide* for more information.

2.2 Control Statement Format

All sort/merge control statements have the same general syntax:

+--- position 1 must be blank unless a label or asterisk(*) is present

Pos.	Pos.	
2	71 72 73	80

-----+

< operation > < operands > < comments > cc < sequence >

The control statements are free-form; the operation definer, operands and comments may appear anywhere between positions 2 and 71. The first position of control statements must contain a blank or a label. Labels, if present, must begin with an alphabetic character in position 1 and may extend up to and including position 8. They must conform to Assembler label format rules.

An asterisk (*) in position 1 denotes a comment.

Any number of continuation statements are permitted.

Operation Field

This field, which identifies the type of statement, must be the first field in a control statement. This field cannot be continued. Operation types (statement definers) recognized by Unicenter CA-Sort are:

SORT
MERGE
SORT1-SORT9
RECORD
MODS
INCLUDE
OMIT
SUM
INREC
OUTREC
ALTSEQ
INPFIL
OUTFIL
OPTION
ANALYZE
END

Each operation type may appear only once.

Operand Field

The operand field must be separated from the operation field by at least one blank character. The operand field consists of one or more operands separated by commas. This field must begin on the first record of the statement. Each operand consists of a keyword (name that identifies the operand type to the sort/merge program) and, optionally, values may be associated with it as follows:

- keyword
- keyword=value
- keyword=(value¹, value², valueⁿ)

When an operand of the type, keyword=(value¹, value², valueⁿ), is used, values may be omitted if they are equal to those assumed by the program. The following rules apply to omitting values from an operand:

- Values may be dropped from right to left. Thus, if all values after value² are equal to those assumed by the program, the operand may be written: keyword=(value¹, value²).
- If values are dropped from the middle or from left to right, commas must be used to signify their omission. Thus, if value² is equal to the value assumed by the program, the operand may be written: keyword=(value¹,,value³).
- If only value¹ of a series is needed, the parentheses are optional keyword=value¹ or keyword=(value¹) may be written.

Comments Field

This field must be separated from the operand field by at least one blank character. It may contain any information.

Continuation Position (72)

Any character other than a blank in this position specifies that the present statement is continued on the next record. This field is not required to be coded if the last character on the statement to be continued is a comma.

Position 73-80

This field is ignored and can be used for any purpose by the user.

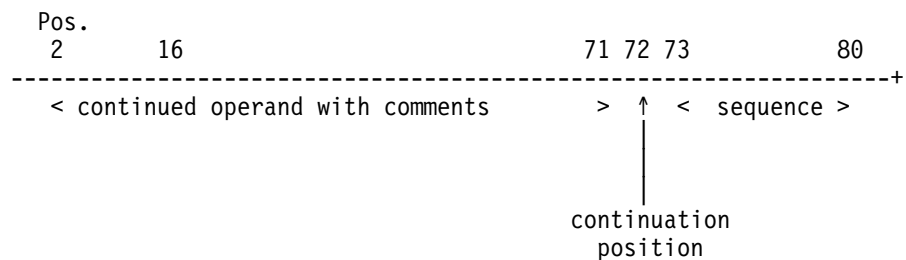
2.3 Continuation Records

A continuation record is treated as a logical extension of the preceding record. An operand or comments field may begin on one record and continue on the next.

If there is continuation of an operand, the next character of the continued operand must appear between positions 2 and 16 of the continuation record.

Additional operands may be placed on subsequent records without filling the first record through position 71 by following a complete operand with a comma and a blank. A nonblank character in the continuation position is optional in this case.

Here, too, the continued operand must begin between positions 2 and 16 of the next statement.



Comments must be continued as follows:

- If the last operand ends in position 71, the comments must begin in position 17 on the continuation record. Otherwise, the comment must begin in position 16.
- A comment statement with an asterisk in position 1 may not be continued. Each comment of this type must have an asterisk in the first position.

Examples of Continuation Statements

		Position													
		1		2		3		4		5		6		7	
		12345678901234567890123456789012345678901234567890123456789012													
:	:	:													
SORT	:	FIELDS=(58,6,D,42,2,A,84,10,A),FORMAT=*													
:	CH,SIZE=15800	COMMENTS *													
:	FURTHER COMMENTS	:													
:	:	:													
:	:	:													
:	SORT	FIELDS=(58,6,D,42,2,													
:	A,84,10,A),FORMAT=CH,SIZE=15800	COMMENTS													
:	:	:													
:	:	:													
:	SORT FIELDS=(58,6,D,42,2,A,84,10,A),FORMAT=CH,	:													
:	SIZE=15800 COMMENTS	:													
:	:	:													
:	:	:													
:	SORT	FIELDS=(58,6,D,42,2,A,84,10,A),FORMAT=CH,SIZE=15800 COM*													
:	MENTS	:													
:	:	:													
:	:	:													
:	SORT	FIELDS=(58,6,D,42,2,A,84,10,A),FORMAT=CH,SIZE=15800 *													
:	COMMENTS	:													
:	:	:													

2.3.1 Notes on Control Statements

- Imbedded blanks are not allowed in values (except within a literal string) or between operands. Everything following a blank is considered part of the comments field.
- Values may consist of up to eight alphanumeric characters.
- Commas, equal signs, parentheses and blanks are field delimiters. They are not allowed as values (except within a literal string).
- Each operation type may appear only once.

2.4 Control Statement Types

This section describes the parameters and options associated with each of the control statements accepted by Unicenter CA-Sort. The SORT (or MERGE) statement is required in all executions. The RECORD control statement is required unless record length information may be retrieved by Unicenter CA-Sort from the Unicenter CA-Dynam interface. All other control statements are optional and may be included if the particular keywords and values are necessary. A summary of these statements follows.

Table Summary of Control Statements

Statement	Operands	Comments
SORT	FIELDS=(f ¹ ,...,fn) each of f ¹ ,...,fn is p,l,r,s FIELDS=(g ¹ ,...,gn),FORMAT={CH BI ZD PD FI FL AQ CSL CST CLO CTO AC Y2C Y2Z Y2S Y2P Y2D PD0 Y2B} each of g ¹ ,...,gn is p,l,s FIELDS=COPY KEY=name WORK={n DA} SIZE=n SIZE=E1 FILES=n {CKPT CHKPT} {EQUALS NOEQUALS} BIAS=	Unicenter CA-Dynam/FI Default=1
MERGE	FIELDS=(f ¹ ,...,fn) each of f ¹ ,...,fn is p,l,r,s FIELDS=(g ¹ ,...,gn),FORMAT={CH BI ZD PD FI FL AQ CSL CST CLO CTO AC Y2C Y2Z Y2S Y2P Y2D PD0 Y2B} each of g ¹ ,...,gn is p,l,s FIELDS=COPY KEY=name {FILES ORDER}= nnnnn	Unicenter CA-Dynam/FI Default=1

Statement	Operands	Comments
SORTx RECORD	FIELDS=(f ¹ ,...,f ⁿ) each of f ¹ ,...,f ⁿ is p,l,r,s FIELDS=(g ¹ ,...,g ⁿ),FORMAT=xx each of g ¹ ,...,g ⁿ is p,l,s FILES=n SYSNO=nnn TYPE={ F V VS } LENGTH=(l ¹ ,l ² ,l ³ ,l ⁴ ,l ⁵) DELBLANK={ (p,q) p }	Default=1 Required param Required param
MODS	PHn=(name,loading info,exit,..)	
INPFIL	BLKSIZE=nnnnn BYPASS SYSIPT {EXIT VSAM [,TOL] ISAM} VOLUME=(n,...) OPEN={RWD NORWD} CLOSE={RWD UNLD NORWD} DATA=E DATA=A [,BUFOFF=nnnn] PRESEQ INPUT SKIPBYTE	Default=1 ¹ Card input Default=1 Default=RWD Default=RWD Default Ignored Ignored Ignored
OUTFIL	BLKSIZE=nnnn {EXIT KSDS ESDS RRDS} REUSE TOL NOTPMK OPEN={RWD NORWD} CLOSE={RWD UNLD NORWD} {SYSLST PUNCH PCHLST} {DUMP CHAR HEX LIST} TITLE='xxxxxxxxxxxxx' CADART SPAN BUFOFF=nnnn BYPASS FREEOUT OUTPUT	Default=1 ³ Exit or VSAM VSAM/VSAM(SAM) VSAM/VSAM(SAM) Default=RWD Default=RWD Unit record Print formats Report title CA-DART users Ignored Ignored Ignored

Statement	Operands	Comments
OPTION	PRINT={ALL NONE CRITICAL} STORAGE={n nK (nK,VIRT)} SORTOUT=n SORTIN=(n ¹ ,...n ⁹) SORTWK=(n ¹ ,...n ⁹) LABEL=(output,in ¹ ,...in ⁹ ,work) NOMIX CASCADE CHECK=x TP SYSNO=nnn OPT={D E} NOTPMK {VERIFY NOVERIFY} {ERASE NOERASE} RESTART CALCAREA DSPSIZE={n MAX} ADDROUT={A D} ROUTE={LST LOG LAL} NOINC {DUMP NODUMP} FILNM=(output,in ¹ ,...in ⁹ ,work) WORKNM=xxxx {PARAMS PARAMS=name} {DIAG NODIAG} DYNAM={Y N} DYNAMFI={Y N} {EQUALS NOEQUALS} PREOPEN={YES NO} SKIPREC=n NRECS=n {CHALT NOCHALT} Y2PAST=n	Default=ALL Default=partition size or value set at install. Default=S Default=Y Default=000 Default set in CASORT macro Default=NOVERIFY Ignored Ignored Ignored - Use ANALYZE stmt Default=SORT Default=SORTPARM Default=set at gen Default=set at gen Default=set at gen Default=set at gen Default=0 Default=all Default=80

2.5 Sort/Merge Control Statements

2.5.1 SORT Control Statement

Specifies the control fields in the input record, as well as the format and sequence to be used when sorting.

```

      { FIELDS=(f1,...f32)
      {   where each of f1,...,f32 is p,l,r,s }
SORT { FIELDS=(g1,...g32),FORMAT=xx           [ ,WORK={ n|DA } ]
      {   where each of g1,...,g32 is p,l,s }
      { FIELDS=COPY
      { KEY=name
      {
      [,SIZE=n ] [,FILES=n]  [{CKPT|CHKPT}]  [{EQUALS|NOEQUAL}]
      [,SIZE=E1 ]
  
```

Field	Description																												
SORT	Statement definer = SORT (sort to be performed)																												
FIELDS=(Keyword for the control-field definition.																												
p	Control-field position relative to the beginning of the logical record. The first byte of a fixed-length record has a relative position of 1. Variable-length data begins in position 5. The maximum value of this parameter is 4092.																												
l	Control-field length in bytes.																												
FIELDS=COPY	Sort function acts as a copy utility. Input files are opened and read in sequence. Less efficient than MERGE FIELDS=COPY.																												
r	Control field data formats:																												
	<table><tr><th>r</th><th>Length in Bytes</th><th>Signed</th><th>Data Format</th></tr><tr><td>CH</td><td>1-256</td><td>NO</td><td>EBCDIC character</td></tr><tr><td>BI</td><td>1-256</td><td>NO</td><td>Binary</td></tr><tr><td>ZD</td><td>1-32</td><td>YES</td><td>Zoned decimal</td></tr><tr><td>PD</td><td>1-32</td><td>YES</td><td>Packed decimal</td></tr><tr><td>FI</td><td>1-32</td><td>YES</td><td>Fixed-point</td></tr><tr><td>FL</td><td>1-256</td><td>YES</td><td>Floating-point</td></tr></table>	r	Length in Bytes	Signed	Data Format	CH	1-256	NO	EBCDIC character	BI	1-256	NO	Binary	ZD	1-32	YES	Zoned decimal	PD	1-32	YES	Packed decimal	FI	1-32	YES	Fixed-point	FL	1-256	YES	Floating-point
r	Length in Bytes	Signed	Data Format																										
CH	1-256	NO	EBCDIC character																										
BI	1-256	NO	Binary																										
ZD	1-32	YES	Zoned decimal																										
PD	1-32	YES	Packed decimal																										
FI	1-32	YES	Fixed-point																										
FL	1-256	YES	Floating-point																										

CSL	2-256	YES	Numeric, separate leading sign
CST	2-256	YES	Numeric, separate trailing sign
CLO	1-256	YES	Numeric, leading overpunch sign
CTO	1-256	YES	Numeric, trailing overpunch sign
AQ	1-256	YES	Alternate collating sequence
AC	1-256	NO	ASCII character
Y2C	2	N/A	2-digit, 2-byte character year data
Y2Z	2	N/A	2-digit, 2-byte zoned decimal year data
Y2S	2	N/A	2-digit, 2-byte binary or zoned decimal year data with the exception that header and trailer records are ignored. Header records contain binary zeroes or blanks in the sort field and trailer records contains X'FFs in the sort field.
Y2P	2	N/A	2-digit, 2-byte packed decimal year data
Y2D	1	N/A	2-digit, 1-byte decimal year data
Y2B	1	N/A	2-digit, 1-byte binary year data
PD0	2-8	N/A	packed decimal with sign and first digit ignored

Field	Description
s	Ordering sequence: A Ascending sequence D Descending sequence
FORMAT=xx	If all control fields are of the same type, the r parameter may be omitted and this keyword used. The xx value should be one of the valid data format types.
WORK=n	Specifies the number of SD or VSAM/SAM DLBL statements supplied in the job stream (number of work files). n may be a value from 0 to 9. If WORK=0 is specified, this indicates that all input will be sorted entirely in-core. DLBL statements supplied in the job stream (if any) will be ignored.
WORK=DA	Indicates that the single work file is a multi-extent DA file. Not allowed for FBA work files. WORK=DA is the same as WORK=1, and Unicenter CA-Sort will determine the actual work file type from the DLBL submitted for SORTWK1 (see notes).
SIZE=n	Optional, estimated number of input records. If ANALYZE CALC is specified, this parameter is required.
SIZE=E1	Optional, load a SORTPARM exit routine to override sort parameter. For details, see the chapter entitled "Defining User Exits."
FILES=n	Optional parameter giving the number of input files. If omitted, n=1 is assumed. The value for n must be in the range of 1-9.
CKPT CHKPT	Specifies that an automatic checkpoint is to be taken at the end of Phase 2.
EQUALS	When control field contents are identical, the input sequence of records is to be retained on output. This value may be set as the default at Unicenter CA-Sort installation time.
NOEQUALS	The default, unless overridden at installation or on the Unicenter CA-Sort OPTION statement, causing equally keyed records to be output in any sequence.
KEY=name	Specifies that the sort/merge keys are to be obtained from the Unicenter CA-Dynam/FI catalog, using the symbolic name 'name'. Unicenter CA-Sort must have been generated with DYNAMFI=Y, and Unicenter CA-Dynam/FI Release 5.0 or higher must be installed and active on the system. The KEY= and FIELDS= parameters are mutually exclusive. This parameter may not be specified when running in a CMS/DOS environment (Unicenter CA-Dynam/FI does not support CMS/DOS).

2.5.2 MERGE Control Statement

Specifies the control fields in the input record, as well as the format and sequence to be used for a merge. This statement is also required if you wish to use sort as a copy utility.

```

      { FIELDS=(f1,...,f32) }
      {   where each of f1,...,f32 is p,l,r,s }
MERGE { FIELDS=(g1,...,g32),FORMAT=xx }      {,FILES=n }
      {   where each of g1,...,g32 is p,l,s }      { }
      { FIELDS=COPY }      {,ORDER=n }
      { KEY=name }      { }

```

Field	Description
MERGE	Statement definer = MERGE (merge to be performed)
FIELDS=(Keyword for the control-field definition.
p l r s	Control-field definition parameters (same format as in the SORT control statement).
FORMAT=xx	If all control fields are of the same type, the parameter may be omitted and this keyword used. The xx value should be one of the valid data format types.
FILES=n	Indicates the number of merge input files. The value for n must be in the range of 1-9. FILES= is required unless FIELDS=COPY is specified. If FIELDS=COPY is specified, the value of n defaults to 1.
ORDER=n	Same meaning as the FILES= parameter.
KEY=name	Specifies that the sort/merge keys are to be obtained from the Unicenter CA-Dynam/FI catalog, using the symbolic name 'name'. Unicenter CA-Sort must have been generated with DYNAMFI=Y, and the Unicenter CA-Dynam/FI Release 5.0 or higher product must be installed and active on the system. The KEY= and FIELDS= parameters are mutually exclusive.
FIELDS=COPY	Merge function acts as a copy utility. All input files are opened at the beginning. More efficient than SORT FIELDS=COPY.

2.5.3 Notes on SORT/MERGE Control Statement

- The collating sequence is defined by up to 32 control fields. Each control field is defined by a group of four parameters (p,l,r,s). The priority of the control fields is given by the sequence of the control-field definitions in the FIELDS= keyword, where the leftmost control field represents the most significant and the rightmost the least significant field.
- If all control field data formats (fn) are identical, they may be omitted and the FORMAT keyword used instead. The FORMAT keyword may not be used if format types are specified in the FIELDS definition as well. The following two control statements are equivalent:

FIELDS=(10,5,CH,A,15,1,CH,A)

FIELDS=(10,5,A,15,1,A),FORMAT=CH

- The relative starting position, p, may not exceed 4092 and all control fields must be completely contained within the record. Variable-length records must have the key contained within the shortest record.
- The sum of all control field lengths, l, in the FIELDS= keyword may not exceed 4092.
- Floating-point control fields must contain normalized arithmetic data (single or double precision).
- Unicenter CA-Sort does not perform packed comparisons on PD or ZD control fields. Therefore, invalid signs in such fields will be accepted. As a result, the sequence of the sorted records may be unpredictable.
- EQUALS is ignored when SUM, OUTREC or ADDROUT is specified and is the default for all MERGE executions.
- The CKPT parameter indicates that a standard VSE checkpoint is to be taken at the end of Phase 2. No checkpoint will be taken for small sorts that do not perform the intermediate merge phase. To continue an interrupted job, the system restart facilities must be used.
- No more than two organizations or device classes (disk, tape, and so forth) may be mixed as input to a merge-only execution at a time. There is no limitation for sorts.
- The KEY= parameter may only be specified if Unicenter CA-Dynam/FI Release 5.0 or higher is installed and Unicenter CA-Sort has been generated with DYNAMFI=Y (yes). Users without Unicenter CA-Dynam, must specify sort/merge keys in the normal fashion.
- The KEY= and FIELDS= parameters are mutually exclusive. In addition, the FORMAT= keyword may not be used if KEY= has been specified.
- If the WORK= parameter is not specified, or WORK=DA is specified, WORK=1 is assumed. Unicenter CA-Sort will then determine from the work file label record (default: SORTWK1 DLBL) the type of DTF to be generated - DA, SD or VSAM/SAM.

- The SORT/MERGE control statement may be used to correctly order existing date fields in various formats. Y2C, Y2Z, Y2S, Y2P, Y2D, and Y2B may be used to order two-digit character, zoned decimal, packed decimal, or decimal year data according to the century window that is set. For example, these formats can be used to order 96, representing the year 1996, before 00, representing the year 2000, in ascending sequence or to order 00 before 96 in descending sequence. In addition, PD0 may be used to order parts of packed decimal fields, such as month and day.

Y2C or Y2Z

May be used to order two-digit, two-byte character or zoned decimal year data. Both Y2C and Y2Z can be specified in the SORT and MERGE statements. They are interchangeable and can be used for any two-digit field that can be represented as *xyxy*, where *y* is hexadecimal 0-9 and represents a year digit, and *x* is hexadecimal 0-F and is ignored.

For example, 96, represented as X'F9F6', X'F9C6', and X'0906', can be handled by Y2C or Y2Z. Thus, you can use Y2C and Y2Z for two-digit years in various types of dates such as C'yy', C'mm/dd/yy', C'yy.mm.dd', Z'ddmmmy', Z'yyddd', C'dddy' and C'yy-mm'.

Y2S You must use Y2S **instead of Y2C or Y2Z** to sort records containing character or zoned decimal two-digit, two-byte year data if the data contains header and or trailer records in the year field. Header records contain binary zeroes or blanks in the year field. Trailer records contain X'FFFF' in the year field.

Y2P May be used to order two-digit, two-byte packed decimal year data. Y2P can be specified in the SORT and MERGE statements and can be used for any two-digit year field that can be represented as *xyyx*, where *y* is hexadecimal 0-9 and represents a year digit, and *x* is hexadecimal 0-F and is ignored.

For example, 96, represented as X'096C', X'096F', and X'8962', can be handled by the Y2P format. Thus, you can use Y2P for two-digit years in various types of dates such as P'yy', P'mmddy', P'yymmdd', P'ddmmmy', P'dddy' and P'mmyy'.

Y2D May be used to order two-digit, one-byte decimal year data. Y2D can be specified in the SORT and MERGE statements. It can be used for any two-digit year field that can be represented as *yy*, where *y* is hexadecimal 0-9 and represents a year digit.

For example, 96, represented as X'96' can be handled by the Y2D format. Thus, you can use Y2D for two-digit years in various types of dates such as X'yy' and P'yyddd'.

Y2B May be used to order two-digit, one-byte binary year data. Y2B can be specified in the SORT and MERGE statements. It can be used for any two-digit year field that can be represented by *yy* where *y* is a hexadecimal 0-F.

For example, year 96, represented as X'60', the binary equivalent for decimal 96, can be handled by the Y2B format.

PD0

May be used to order a packed decimal with sign and first digit ignored. PD0 can be specified in the SORT and MERGE statements. It can be used for any two- to eight-byte packed decimal field you want to treat as *xddd...ds*, where *x* is hexadecimal 0-F and is ignored, *d* is hexadecimal 0-9 and represents a decimal digit, and *s* is hexadecimal 0-F and is ignored.

For example, 12, represented as X'012C', X'012D', X'912C', and X'3127', can be handled by PD0, whereas with the PD format, these values would be treated as +12, -12, +912 and -312, respectively. Thus, you can use PD0 for parts of PD fields such as mm, dd, mmdd and ddmm in 'Pmmddy', 'Pddmmy', 'P'yymmdd' and 'P'yymm'.

2.5.3.1 Specifying Bit Control Fields

Control fields that are defined as type BI (binary values) may be described in bytes and bits as follows:

The relative position, pn, of a BI control field may be defined 'byte.bit', where 'byte' specifies the byte location relative to the beginning of the record and where 'bit' specifies the bit location relative to the beginning of the byte. Note that the first bit of a byte is bit 0, the last bit is bit 7. The length (mn) of a BI control field may also be defined 'byte.bit'.

Example

```
FIELDS=(5.0,1.4,BI,A)
```

This control field starts at the beginning (high-order bit) of the fifth byte of a record and has a length of one byte and four bits.

2.5.3.2 The Merge Copy Utility

Using an extension of the MERGE control statement, Unicenter CA-Sort provides an additional feature that enables the user to create an identical copy of one or more files. This function is achieved by specifying FIELDS=COPY on the MERGE statement. During this merge operation, the input files (up to nine), which need not be presorted, are transferred to another file. Any combination of tape/disk devices may be assigned. All logical records are copied from the input files (SORTIN1, SORTIN2, SORTIN3,...) to the output file (SORTOUT). All input files must be mounted at the same time because they are all opened at the start of the MERGE run. The record selection statements INCLUDE/OMIT, OUTREC or DELBLANK may be used in conjunction with this utility.

Example

```
MERGE  FIELDS=COPY,ORDER=3
RECORD TYPE=F,LENGTH=100,DELBLANK=(51,9)
INPFIL  BLKSIZE=3200
OUTFIL  BLKSIZE=6000
OPTION  LABEL=(U,S,U,S)
```

In this example, three files are required to be copied. The logical records to be selected are those without '9' in position 51 of each record.

2.5.4 SORTx Control Statement

Allows multiple output files in different sorted sequences to be produced from the same input file.

```

SORTx  { FIELDS=(f1,...f20)
        {   where each of f1,...,f20 is p,l,r,s
        {
        { FIELDS=(g1,...g20),FORMAT=f
        {   where each of g1,...,g20 is p,l,s
        {
        [ ,FILES=n ]           [ ,SYSNO=nnn ]

```

Field	Description																														
SORTx	Operation type=SORTx, where x=1-9.																														
FIELDS=(Keyword of the definition of control fields.																														
p	Relative control field position in each logical record. The first byte of a fixed-length record has relative position 1; the first byte of a variable-length record has relative position 5. Record has relative position 5. The maximum value for p is 4092.																														
l	Length of the control field in bytes.																														
s	Ordering sequence: A Ascending sequence D Descending sequence																														
r	Control field data format: <table><tr><td>f¹⁻²⁰</td><td>Length</td><td>Contents</td></tr><tr><td>CH</td><td>1-256</td><td>EBCDIC Character</td></tr><tr><td>BI</td><td>1-256</td><td>Unsigned Binary</td></tr><tr><td>PD</td><td>1-32</td><td>Packed Decimal</td></tr><tr><td>Y2C</td><td>2 bytes</td><td>2-digit, 2-byte character year data</td></tr><tr><td>Y2Z</td><td>2 bytes</td><td>2-digit, 2-byte zoned decimal year data</td></tr><tr><td>Y2S</td><td>2 bytes</td><td>Similar to Y2C and Y2Z when the data contains header and or trailer records in the year data.</td></tr><tr><td>Y2P</td><td>2 bytes</td><td>2-digit, 2-byte packed decimal year data</td></tr><tr><td>Y2D</td><td>1 byte</td><td>2-digit, 1-byte decimal year data</td></tr><tr><td>Y2B</td><td>1 byte</td><td>2-digit, 1-byte binary year data</td></tr></table>	f ¹⁻²⁰	Length	Contents	CH	1-256	EBCDIC Character	BI	1-256	Unsigned Binary	PD	1-32	Packed Decimal	Y2C	2 bytes	2-digit, 2-byte character year data	Y2Z	2 bytes	2-digit, 2-byte zoned decimal year data	Y2S	2 bytes	Similar to Y2C and Y2Z when the data contains header and or trailer records in the year data.	Y2P	2 bytes	2-digit, 2-byte packed decimal year data	Y2D	1 byte	2-digit, 1-byte decimal year data	Y2B	1 byte	2-digit, 1-byte binary year data
f ¹⁻²⁰	Length	Contents																													
CH	1-256	EBCDIC Character																													
BI	1-256	Unsigned Binary																													
PD	1-32	Packed Decimal																													
Y2C	2 bytes	2-digit, 2-byte character year data																													
Y2Z	2 bytes	2-digit, 2-byte zoned decimal year data																													
Y2S	2 bytes	Similar to Y2C and Y2Z when the data contains header and or trailer records in the year data.																													
Y2P	2 bytes	2-digit, 2-byte packed decimal year data																													
Y2D	1 byte	2-digit, 1-byte decimal year data																													
Y2B	1 byte	2-digit, 1-byte binary year data																													
FORMAT=f	If all control fields are the same data format type, the parameter r may be omitted and FORMAT=f coded instead.																														
FILES=n	The largest values specified on all SORTx statements submitted will be used to indicate the number of input files to be read by Unicenter CA-Sort.																														

Field	Description
SYSNO=nnn	<p>The SYSNO keyword is used to override the default logical unit to be used for the SORTOTn file. The default logical units for sorted output files are SYS020 for SORTOT1, SYS021 for SORTOT2,..., SYS028 for SORTOT9.</p> <p>If this parameter is placed on one SORTx statement, it must be specified for all SORTx control statements in the run.</p>

2.5.4.1 Multiple Output Files - Overview

Through the use of Unicenter CA-SRAM, Computer Associates Sort Reentrant Access Method, the SORTx control statement allows up to nine separate sorted output files to be produced from a single execution of the sort/merge package. The input data sets will be read only once, regardless of the number of output files, thus potentially saving a substantial amount of I/O operations. Each output file may be ordered in a different sequence, using control field definitions similar to those used with the SORT or MERGE control statements.

As a default, the sorted output files are created simultaneously during the sort execution by dividing the available main storage in the partition among the 1-9 Unicenter CA-SRAM sorts. When using this parallel sorting technique, a disk work file is required for each concurrent sort.

When the number of input records is extremely large, or the available storage in the partition is limited, a cascading sort technique may be chosen by placing the keyword CASCADE on the OPTION control statement. When this method is chosen, Unicenter CA-Sort divides the available main storage between only two sorts at a time. The output of the first sort becomes input to the second, the output of the second is cascaded as input to the third, and so on, until all the SORTx statements submitted have been processed. Only two work files are required when this sorting technique is used. The CASCADE option should be preferred for applications which run in small partitions or are producing a large number of sorted output files.

For more information on the use of Unicenter CA-SRAM, and the parallel and cascading sort techniques described above, refer to the *Unicenter CA-SRAM User Guide*.

2.5.4.2 SORTx Specifications

- The SORTx statements must begin with SORT1 and continue in consecutive order (for example, SORT1, SORT3, SORT4 is invalid). To sort to a single output file, either the SORT or SORT1 control statement may be used.
- Unlike when the SORT control statement is used, work file logical units must be assigned to the proper disk device type during Phase 0 if SORTx statements are used.
- The keyword FILES=n is accepted on the SORTx statement. The largest value found on any SORTx statement is considered to be the number of input files. Input file DTF names remain SORTIN1 through SORTIN9.
- Work files used by the Multiple Output File Facility use the file name SORT1 for the SORT1 control statement and SORT2 for the SORT2 SORTx control statement, unless the CASCADE option is specified on the OPTION statement. When using CASCADE, only the work files SORT1 and SORT2 are required by Unicenter CA-Sort.
- Likewise, there must be a corresponding output file for every SORTx statement used. The output file used for SORT1 must have the DTF name SORTOT1, the output file used for SORT2 must have the DTF name SORTOT2, and so forth.
- The 1-9 output files must all be tape, all CKD disk, all FBA disk, or all VSAM. The mixing of organizations or device classes is not permitted. However, the output files may be mixed within device classes (for example, SORTOT1 may be allocated on a 3330 CKD disk and SORTOT2 may reside on a 3350 CKD disk).
- The formats for the 1-20 control fields must be either BI, CH, or PD. No other format types are supported.

The following may be used with SORTx:

- E15 user exits
- INCLUDE/OMIT control statement
- OUTREC control statement
- FILNM keyword of the OPTION control statement
- WORKNM keyword of the OPTION control statement

The following may **not** be used in conjunction with SORTx:

- SORT or MERGE control statement
- SUM control statement
- ALTSEQ control statement
- EQUALS option
- Phase 3 user exits
- Bit control fields
- Automatic in-core sort

The following will be ignored when using SORTx:

- Checkpoints will be ignored when using the Multiple Sorted Output File Facility.
- The parameters WORK, SIZE, BIAS, CKPT, CHKPT, EQUALS, and NOEQUALS as found on the SORT and MERGE control statements will be ignored if specified on any SORTx statement.
- The SORTOUT keyword of the OPTION control statement will be ignored. Instead, the keyword SYSNO=nnn may be added to the SORTx control statements to override the output logical unit.

2.5.5 RECORD Control Statement

Specifies the record length and record type of the input file.

```

RECORD TYPE= { F } , LENGTH= { (I1, I2, I3, I4, I5) }
              { V }           { (I1, I2, I3, I4) }
              { VS }          { (I1, I2, I3) }
                                { (I1, I2) }
                                { I1 }
                                [ , DELBLANK= { (p,q) }
                                           { p } ]

```

Field	Description
RECORD	Statement definer = RECORD
TYPE=	Record type (required) <ul style="list-style-type: none"> F Fixed-length blocked or unblocked V Variable-length blocked or unblocked VS Variable spanned blocked
LENGTH = (Record length definition <ul style="list-style-type: none"> I¹ Length of records in the input file (required). I² Internal length to be used by the sort (for example, after the record length modification in the exit routine E15). Defaults to I¹ if not specified. I³ Length of record in the output file if record length modification is performed by a user exit routine in the final merge phase at exit E35. I⁴ Minimum record length for the variable-length record in the input file, or after exit E15. I⁴ defaults to the greater of either 18 or the minimum record size necessary to contain the control fields specified in the SORT statement. I⁵ Modal record length for variable-length records. Defaults to the average of I² and I⁴.
DELBLANK=	Delete record option (optional)
(p,q) p	<ul style="list-style-type: none"> p Position in the logical record q Character

2.5.5.1 Notes on the LENGTH Operand

- The LENGTH operand for fixed-length records has the format:

$$\text{LENGTH} = (l^1, l^2, l^3)$$

- The LENGTH operand for variable-length records has the format:

$$\text{LENGTH} = (l^1, l^2, l^3, l^4, l^5)$$

- In all LENGTH definitions only the l^1 value is required. Other values are optional and may be dropped, replacing them with commas.
- The l^1, l^2, l^3 values for variable-length records designate the maximum length limits and may not exceed the values shown in the previous table for the particular devices.
- The default value for minimum logical record length is 12 bytes. If the value l^4 is not specified, it is assumed to be the greater of either 12 bytes (tape files only) or the greatest value consistent with the specification given in the SORT, MERGE, SUM, INREC, OUTREC, INCLUDE or OMIT control statements.
- The values l^2 and l^4 are not used for MERGE operations.
- The l^5 parameter specifies the record length that occurs most frequently in the input file (modal length). If a value for l^5 is not specified, it is assumed to be equal to the average of the maximum and minimum record lengths in the input file, for example, Unicenter CA-Sort assumes:

$$l^5 = \frac{l^4 + l^2}{2}$$

Maximum assumed $l^5=816$ bytes. The average record length for a given sort, displayed by the message CASD214I, may be used as the l^5 value for subsequent sorts of similar sort input files.

- The lengths specified for variable-length records must include the 4-byte count field that the disk operating system standards require as a Record Descriptor Word. The Block Descriptor Word in VB files is removed by Unicenter CA-Sort.
- If either input or output files are allocated to a tape unit, the minimum physical record length of the file(s) is 12 bytes. If both input and output files are allocated to DASD, the minimum record length is 1 byte.

2.5.5.2 Notes on Using Optional Keywords

- If the ADDROUT or ADDROUT=A option is specified on the OPTION statement, l² must be equal to 10 (bytes) plus the sum of the individual control field lengths for sequential files, or 5 (bytes) plus the sum of the control field lengths for VSAM files. This value is used by default if you do not specify a value for l².
- When ADDROUT=A is specified on the OPTION statement, l³ must be equal to 10 when processing sequential files, or 5 when VSAM files are being processed. When ADDROUT=D is specified, l³ must be equal to 10 plus the sum of the individual control field lengths. These values are used by default if the ADDROUT option is used and you do not specify a value for l³.
- ADDROUT=D is not allowed for VSAM files.
- For compatibility with the IBM Sort/Merge, the SPAN keyword may be placed on the OUTFIL control statement to indicate variable-spanned output is to be produced, instead of specifying TYPE=VS on the RECORD control statement.
- If both p and q are specified in the DELBLANK operand, all records with character q at position p are omitted. If only p is specified, all records containing a blank at position p are omitted.

q may not take any of the following values:

() , =

(left parenthesis, right parenthesis, comma, blank, equal sign)

RECORD Statement Examples

```
RECORD LENGTH=(80,,60),TYPE=F
```

```
RECORD LENGTH=(300,180,185,60,120),TYPE=V
```


2.5.6 MODS Control Statement

Specifies the user exits and other required information when exits are to be invoked.

```
MODS PHn=(name,loading-information,exit,exit,...), ...
```

Field	Description
MODS	Statement definer = MODS (user exit definition)
PHn=(<p>The value n specifies the sort/merge program phase which is to be modified by the user's exit routines. The acceptable keywords are:</p> <p>PH1 Internal sort phase</p> <p>PH2 Intermediate merge phase</p> <p>PH3 Final merge phase (or merge-only run)</p> <p>All three keywords may be specified.</p>
name	Cataloged name of the exit module in the core image library. The name must be omitted if the associated exit routines are in main storage (preloaded) when Unicenter CA-Sort is invoked.
loading information	<p>Specifies either:</p> <ul style="list-style-type: none">■ The length of the exit routine (decimal value prefixed by the letter 'L') if the user's exit is relocatable (for example, can be automatically relocated by the system loader). Unicenter CA-Sort will load these routines at the most suitable main storage address.■ The absolute loading address (in decimal, on a doubleword boundary) for nonrelocatable exit routines. Unicenter CA-Sort will load the particular exit routine at the specified address and not use any storage above this address. <p>The loading information must be omitted if the exit routines are already in main storage (for example, the sort/merge has been initiated by the LOAD macro).</p>
exit	Specifies the sort/merge exits to be used in the particular program phase.

Because of the storage management techniques used by Unicenter CA-Sort, it is very important that the Lnnnn parameter for self-relocating exits be specified correctly. Unicenter CA-Sort will reserve storage for the user routine based upon the value specified in the length parameter of the MODS statement, or the actual length of the user routine as determined from the core image library directory entry for the phase, whichever value is larger. If the user routine will require additional storage beyond its own phase length (for example, loading other phases or building tables), the length on the MODS statement must take this into account in order to avoid Unicenter CA-Sort overlaying the user exit storage with its own modules. If Unicenter CA-Sort overlays the user exit storage with its own modules, it will cause unpredictable results.

User exits which load phases into the partition during execution must load these routines or tables immediately after the exit routine. The Lnnnn parameter of the MODS statement must be used to reserve sufficient storage. The 'address of last phase fetched or loaded' field in the partition communication region must NOT be used to locate where a user routine may be dynamically loaded because all storage in the partition, except that reserved using the MODS statement, is in use by Unicenter CA-Sort.

2.5.6.1 MODS Statement Examples

```
MODS PH1=(USERA,19000,E11,E17)
```

Above, routines contained in the phase USERA are used at exits E11 and E17 of Phase 1 of the sort/merge program, probably to handle nonstandard tape labels. The routines in USERA are to be loaded at decimal location 19,000 in main storage (not relocatable).

```
MODS PH2=(USERB,L600,E25),PH3=(USERC,L700,E31,E38,E39)
```

Here, exits are specified for the second and third phases of the sort/merge program. Phase USERB contains a self-relocating or relocatable routine no more than 600 bytes long, which will be executed at exit E25. USERC is 700 bytes long and contains the relocatable routines executed at exits E31, E38 and E39. Relocatable exit routines are recommended to allow Unicenter CA-Sort to utilize storage most efficiently.

```
MODS PH3=(,E31,E35)
```

Two user exits are specified above for the third execution phase. The exit routine is already in main storage, so it is unnecessary to include its name and loading specifications. The absence of these values is indicated by commas. Preloaded user routines are used only for sorts and merges that are initiated from another program.

2.5.7 INPFIL Control Statement

The INPFIL control statement is required if the input file is blocked or if tape volume information must be passed.

```

                                [ BLKSIZE=n ] [ ,OPEN= { RWD | NORWD } ] [ ,CLOSE= { RWD }
                                                                { UNLD } ]
                                                                { NORWD } ]
INPFIL [ ,VOLUME=n ] [ ,BYPASS ]
        [ ,EXIT ] [ ,ISAM ] [ ,VSAM ] [ ,TOL ] [ ,SYSIPT ]
        [ ,DATA=E ] [ ,DATA=A [ ,BUFOFF=nnnn ] ]
        [ ,PRESEQ ] [ ,INPUT ] [ ,SKIPBYTE ]

```

Field	Description
INPFIL	Statement definer = INPFIL (input file definition)
BLKSIZE=n	The value "n" specifies the maximum block size in bytes for input file. The default value is equal to the l ¹ parameter in the RECORD statement for fixed-length records and l ¹ + 4 for variable-length records. The block size must be the largest if multiple files will be input to a sort or merge. Block size must include the key length if KEYLEN has been specified on the OPTION control statement.
BYPASS	Incorrectly read tape input blocks will be dropped. Message CASD204E will print at the end of Phase 1. BYPASS is ignored if exits E18 and E38 are active.
EXIT	Specifies that a user exit routine (E15 for a sort, E32 for a merge) will supply all input records. The SORTINn files will be ignored. All other INPFIL parameters are ignored, except SYSIPT.
VOLUME=	This operand specifies the number of tape input volumes in the format VOLUME=(n ¹ ,n ² ,n ³ , ... n ⁹) where n ¹ -n ⁹ represents the counts for the SORTIN1 through SORTIN9 files, respectively. The default is one volume for unlabeled files. The VOLUME parameter is ignored for standard labeled files.
VSAM	Input files are native VSAM (KSDS, ESDS or RRDS). If VSAM is not specified, input is assumed to be SAM or VSAM/SAM. VSAM-managed files may be accessed if VSAM is specified, but native VSAM and VSAM/SAM files may not be mixed if VSAM keyword is omitted.
ISAM	Input is an indexed sequential file with fixed-length records. The BLKSIZE parameter must be specified if the file is blocked.

Field	Description
OPEN	Specifies the file positioning for the first volume of each tape input file before processing starts. RWD The tape is rewound (the default). NORWD The tape will not be rewound.
CLOSE	Specifies tape file positioning at end-of-file. RWD The tape is rewound (the default). NORWD The tape will not be rewound. UNLD The tape will be unloaded (unload is assumed if no alternate drive for a multi-volume file is available).
TOL	Unicenter CA-Sort will accept a warning code from VSAM and continue processing. Fatal codes will terminate.
SYSIPT	Indicates that input is to be read from SYSIPT. Records are assumed to be 80- or 81-byte records. If the BLKSIZE parameter has been specified, the SORTINn files will also be read, allowing statement input along with the normal input files; otherwise, the SORTINn files will be ignored. User exits at E15 are still allowed when SYSIPT is specified.
DATA=E	Indicates that input is in EBCDIC format (default).
DATA=A	Specifies that the input data is in ASCII format from a 9-track tape unit.
BUFOFF=n	May only be used if DATA=A has been specified (for example, with ASCII input) to indicate the length of the block prefix at the beginning of each physical record. If not specified, BUFOFF=0 is assumed.
PRESEQ SKIPBYTE INPUT	For compatibility with early IBM sort/merge products, these keywords will be accepted, but will be ignored by Unicenter CA-Sort.

2.5.7.1 INPFIL Control Statement Notes

- If ISAM is specified and the file is unblocked, only the DATA portion of the record is made available. Hence, if the ISAM key is not within the record, it will be unavailable to the sort. For unblocked ISAM files, the BLKSIZE= parameter should be omitted.
- If SYSIPT has been specified, user exits may not issue return code 12 to insert records. All other E15 return codes are allowed.
- VSAM-managed SAM files will be read using the native VSAM access method if VSAM is specified on the INPFIL statement. Although the result is the same, this is not the same as using SAM simulation under VSAM (for example, VSAM/SAM). Unicenter CA-Sort will use the VSAM/SAM technique if VSAM has been specified on the DLBL for the input file, and VSAM has not been specified on the INPFIL statement (for example, only BLKSIZE has been indicated). A native VSAM ESDS must be read using VSAM.
- Care should be taken with INPFIL/OUTFIL OPEN and CLOSE tape positioning parameters used in conjunction with tape management products since tape management interception may occur before the OPEN/CLOSE SVC has processed the requested tape DTF positioning.

INPFIL Statement Examples

```
INPFIL  BLKSIZE=1200,OPEN=NORWD,CLOSE=UNLD
INPFIL  SYSIPT,BLKSIZE=80
INPFIL  VSAM,TOL
INPFIL  EXIT
```

2.5.8 OUTFIL Control Statement

The OUTFIL control statement is used to specify numerous optional output parameters: block size, rewinding instructions, routing options, format specifications, etc.

```

                                [ BLKSIZE=n ] [ ,OPEN= { RWD | NORWD } ] [ ,CLOSE= { RWD
                                                                { UNLD } ]
                                                                { NORWD } ]
OUTFIL [ ,NOTPMK ] [ ,EXIT ] [ ,CADART ]
        [ { ,ESDS | KSDS | RRDS } ] [ ,TOL ] [ ,REUSE ]
        [ { ,SYSLST | PUNCH | PCHLST } ] [ ,SPAN ]
        [ { ,DUMP | CHAR | HEX | LIST } ] [ ,TITLE='xxxxxxxxxxxxx' ]
        [ ,BUFOFF=nnn ] [ ,FREEOUT ] [ ,BYPASS ] [ ,OUTPUT ]

```

Field	Description
OUTFIL	Statement definer = OUTFIL (output file definition)
BLKSIZE=n	The value "n" specifies the maximum block size in bytes for the output file. The default value is equal to the l ³ parameter in the RECORD statement for fixed-length records and l ³ + 4 for variable-length records. The block size must include the key length if KEYLEN has been specified on the OPTION Control Statement.
NOTPMK	Specifies that no tape mark is to be written before the first data record on each volume of unlabeled tape output.
EXIT	Specifies that a user E35 exit routine will write the output file. The SORTOUT file will be ignored. Other OUTFIL organization parameters are ignored if EXIT is specified.
ESDS	Output is an entry-sequenced, native VSAM file. All other parameters except TOL and REUSE are ignored.
KSDS	Output is a key-sequenced, native VSAM file. All other parameters except TOL and REUSE are ignored.
RRDS	Output is a relative-record, native VSAM file. All other parameters except TOL and REUSE are ignored.
TOL	Unicenter CA-Sort will accept a warning code from VSAM and continue processing.
REUSE	If an existing VSAM file is defined as reusable, the file will be treated as empty when opened.
OPEN	Specifies the file positioning for the first volume of each tape output file before processing starts. RWD The tape is rewound (the default). NORWD The tape will not be rewound.

Field	Description
CLOSE	Specifies tape file positioning at end-of-file. RWD The tape is rewound (the default). NORWD The tape will not be rewound. UNLD The tape will be unloaded (unload is assumed if no alternate drive for a multi-volume file is available).
CADART	Specifies that CA-DART is to be invoked to print a report using CA-DART control statements submitted after the END control statement. CA-DART must be installed on the system to use this option.
SYSLST	Specifies the sorted records are to be listed on SYSLST. If BLKSIZE= has been specified, they will be listed in addition to being written to SORTOUT; otherwise, only SYSLST output will be produced. DUMP format will be assumed unless overridden. User exits at E35 may be used with SYSLST.
PUNCH	Same as SYSLST, except records will be directed to SYSPCH (and SORTOUT if BLKSIZE= is specified). Records will be truncated to 80 bytes if longer.
PCHLST	Directs output records to both SYSLST and SYSPCH.
DUMP	Prints output records on SYSLST in both character and hexadecimal format (the default).
CHAR	Prints output records on SYSLST in character format.
HEX	Prints output records on SYSLST in hexadecimal only.
LIST	Prints output records on SYSLST in character format without position spacers. When using DUMP, CHAR, HEX, and LIST either SYSLST or PCHLST must be specified.
TITLE	Sets a custom title for output on SYSLST. The title must be enclosed in apostrophes.
BUFOFF=n	If output is to tape and in ASCII format, the BUFOFF value indicates the length of the block prefix which is to be appended to the front of each physical record before it is written by Unicenter CA-Sort. BUFOFF=0 is assumed if the keyword is not specified.
FREEOUT BYPASS OUTPUT	For compatibility with early IBM sort/merge products, these keywords will be accepted, but will be ignored by Unicenter CA-Sort.

Field	Description
SPAN	For compatibility with the IBM sort/merge product, variable-spanned output may be selected by using this keyword instead of specifying TYPE=VS on the RECORD control statement.

OUTFIL Control Statement Examples

```

OUTFIL  BLKSIZE=1200,OPEN=NORWD,CLOSE=UNLD
OUTFIL  SYSLST,CHAR,TITLE='INPUT TRANSACTION RECORDS'
OUTFIL  PCHLST
OUTFIL  CADART
OUTFIL  KSDS,TOL
OUTFIL  EXIT

```

2.5.8.1 OUTFIL Control Statement Notes

- The TOL and REUSE parameters may only be used if the output is a VSAM file (ESDS, KSDS, RRDS or VSAM/SAM).
- If the various VSAM keywords are omitted from the OUTFIL statement, the output is assumed to be SAM, DAM or VSAM/SAM.
- Unicenter CA-Sort will always attempt to block the output file based upon the values specified for BLKSIZE= (or 13 if omitted). It is not needed if the output is VSAM or accessed as VSAM, nor if the output is to be unblocked.
- Likewise, with FBA devices, Unicenter CA-Sort will fill each output control interval to capacity with blocks of maximum size and smaller blocks if space remains in the CI. This is done to conserve DASD space. LIOCS will always accept short blocks as input to another program if the blocks are consistent with the record format (for example, short fixed-length blocks will always divide evenly by the output record length, and short variable-length blocks will always contain complete records).
- User E35 exits may be used when SYSLST, PUNCH and PCHLST have been specified to examine, modify or delete the records before printing. However, the user exit may not insert records using return code 12.
- The NRECS and SKIPREC options on the OPTION control statement may be useful for selecting and limiting records sorted and printed using the SYSLST/PUNCH/PCHLST parameter.
- If the OUTFIL option of CA-DART is to be used to allow SORTOUT to be produced in addition to a report, a BLKSIZE= parameter must be present along with the CADART parameter on the OUTFIL statement.
- The CADART and SYSLST parameters are mutually exclusive.
- Care should be taken with INPFIL/OUTFIL OPEN and CLOSE tape positioning parameters used in conjunction with tape management products since tape management interception may occur before the OPEN/CLOSE SVC has processed the requested tape DTF positioning.

2.5.9 INCLUDE/OMIT Control Statement

Used when either selecting or rejecting certain records from the input file before sorting actually takes place. The use of INCLUDE and OMIT are mutually exclusive.

```
INCLUDE
      COND=c [,FORMAT=r]
OMIT
```

where a condition **c** is of the form

{	(p,l,r)	{,EQ}	{p ² ,l ² ,r ² }	}
{		{,NE}	{constant}	}
{		{,GT}		}
{		{,GE}		}
{		{,LT}		}
{		{,LE}		}
{				}
{		{,AND,}		}
{	(c ¹	{,&,}	c ²)	}
{		{,OR,}		}
{		{, ,}		}
{				}

c¹,c² are nested conditions with up to 8 nesting levels.

Field	Description
INCLUDE	Statement definer = INCLUDE (select records)
OMIT	Statement definer = OMIT (reject records)
COND=(Keyword for the relational expression definition
p	Position of the data field relative to the beginning of the record. The first byte of a fixed-length record has a relative position of 1. The first data byte of a variable-length record has a relative position of 5. The maximum value which may be specified is 4092.
l	Length of the field (must be a whole number of bytes - no bit fields allowed). Maximum length is 256 bytes for character fields. Zoned decimal and packed decimal field lengths must not exceed 16 bytes and fixed-point fields are restricted to 4 bytes.
r	<p>Data Field Format:</p> <p>fn Data Format</p> <p>CH EBCDIC character (no bit fields)</p> <p>BI EBCDIC character (identical to CH format)</p> <p>ZD Signed zoned decimal data</p> <p>PD Packed decimal data</p> <p>FI Fixed-point data</p>
EQ	"Equal" relational operator
NE	"Not Equal" relational operator
GT	"Greater than" relational operator
GE	"Greater than or equal to" relational operator
LT	"Less than" relational operator
LE	"Less than or equal to" relational operator
AND or &	"AND" logical operator
OR or	"OR" logical operator
FORMAT=r	If all data fields are of same type, the fn values may be omitted and FORMAT=r used. The r will then be replaced by the common fn value.

Field	Description
constant	a) decimal number
	\pm nnn where n represents a decimal digit. No decimal point is allowed.
	Decimal constant may be compared only with the ZD, PD or FI data fields. The value may not exceed $\pm 2,147,483,648$ for fixed-point (FI) fields, and 16 digits for ZD and PD fields (see 'INCLUDE/OMIT Statement Notes').
	b) character or hexadecimal constant
	C'xxxx' where x represents any EBCDIC character. If an apostrophe (') is to be within the character string, it must be specified as double apostrophes.
	X'hhhh' where h represents any valid hexadecimal digit 0-9 or A-F. An even number of digits must be specified.

2.5.9.1 INCLUDE/OMIT Statement Notes

- Only one INCLUDE or OMIT or DELBLANK definition may be specified in a single sort/merge execution (INCLUDE, OMIT and DELBLANK are mutually exclusive).
- The logical expression may contain up to 8 levels of nested parentheses to allow a clearer definition of the selection criteria to be specified. By contrast, the IBM sort/merge program allows only a single level of parentheses.
- Character or binary (CH,BI) fields may be compared only with CH,BI fields or C,X-type constants.
- Zoned decimal (ZD) or packed decimal (PD) fields may be compared only with ZD or PD fields (combinations of ZD,PD fields are allowed).
- Fixed-point fields (FI) may be compared only with other fixed-point fields or numeric constants.
- Maximum size of the generated selection routine is limited to 8K. On the average, each relational expression requires approximately 32 bytes. All lengths of the constants and literals must also be added to the generated routine length.
- Character or hexadecimal literal strings are padded on the right with blanks (X'40') or on the left with zeros (X'00'), respectively, if the data length is greater than the length of the constant. Decimal constants are padded on the left with numeric zeros.

INCLUDE/OMIT Statement Examples

```
INCLUDE COND=(2,3,CH,EQ,C'MAY',AND,(9,5,ZD,LT,1000,OR,9,5,ZD,GT,2000))
```

All records containing in positions 2,3,4 the text 'MAY' and where the amount starting in position 9 is either less than 1000 or greater than 2000 will be processed by Unicenter CA-Sort. All other records will be eliminated.

```
OMIT COND=(1,1,BI,EQ,X'FF',OR,3,2,CH,GT,10,2,CH)
```

All records which contain X'FF' in the first byte or where the data in positions 3 through 4 is larger in value than the data in positions 10 through 11 will be excluded from the sort/merge process.

2.5.10 INCLUDEx/OMITx Control Statement

Similar to the INCLUDE/OMIT control statement. Used when selecting or rejecting records in multiple output sorts for the file SORTOTx. The value of x can be from 1 to 9.

See the INCLUDE/OMIT control statement for a description of the syntax.

2.5.11 INREC/OUTREC Control Statement

INREC is used if the input record is to be altered before it goes into the sort. OUTREC is used if the record coming out of the sort is to be altered before being written to the output file.

```
INREC
      FIELDS=(f1,...,fn [,v])
OUTREC
```

where each of f1,...,fn is of the form

$$\begin{Bmatrix} [c:]p,l[,a] \\ f \end{Bmatrix}$$

Field	Description
INREC	Statement definer = INREC (reformat in Phase 1)
OUTREC	Statement definer = OUTREC (reformat in Phase 3)
FIELDS=(Keyword defining the particular record fields and their alignment.
c:	An optional prefix designating the position of the field in the output record. The position must follow the end of the previous field. The gap from the end of the last field is filled in with blanks.

Field	Description																												
p	Position of the data field relative to the start of the input record (after E15 for INREC, and before E35 for OUTREC). The first byte of a fixed-length record has a relative position of 1. The first data byte of a variable-length record has a relative position of 5. The maximum value for p is 32767.																												
l	Length of the field in bytes. Bit lengths may not be specified. The maximum value for l is 32767.																												
a	<p>Alignment of the data in the reformatted record or reformatting of the date field. To specify the alignment of data in the reformatted record, choose one of the following codes:</p> <table> <tr> <th>Code</th><th>Data Alignment On</th></tr> <tr> <td>H</td><td>Halfword boundary</td></tr> <tr> <td>F</td><td>Fullword boundary</td></tr> <tr> <td>D</td><td>Doubleword boundary</td></tr> </table> <p>If this parameter is omitted, alignment occurs to the next byte boundary. Unused space will be inserted as binary zeros (X'00'). The alignment performed is relative to the beginning of the reformatted record. The c: prefix may coexist with the alignment operator in which case the prefix is processed first. To reformat a date field, choose one of the following formats:</p> <table> <tr> <th>Format</th><th>Data Size/Type</th></tr> <tr> <td>Y2C</td><td>2-digit, 2-byte character year data</td></tr> <tr> <td>Y2Z</td><td>2-digit, 2-byte zoned decimal year data</td></tr> <tr> <td>Y2S</td><td>2-digit, 2-byte character year or zoned decimal year data. Used when the data contains header and or trailer records.</td></tr> <tr> <td>Y2P</td><td>2-digit, 2-byte packed decimal year data</td></tr> <tr> <td>Y2D</td><td>2-digit, 1-byte decimal year data</td></tr> <tr> <td>Y2B</td><td>2-digit, 1-byte binary year data</td></tr> <tr> <td>PZ</td><td>2-8 byte packed decimal with sign and first digit ignored</td></tr> <tr> <td>PSI</td><td>1-8 byte packed decimal with sign ignored</td></tr> <tr> <td>ZSI</td><td>1-8 byte zoned decimal with sign ignored</td></tr> </table>	Code	Data Alignment On	H	Halfword boundary	F	Fullword boundary	D	Doubleword boundary	Format	Data Size/Type	Y2C	2-digit, 2-byte character year data	Y2Z	2-digit, 2-byte zoned decimal year data	Y2S	2-digit, 2-byte character year or zoned decimal year data. Used when the data contains header and or trailer records.	Y2P	2-digit, 2-byte packed decimal year data	Y2D	2-digit, 1-byte decimal year data	Y2B	2-digit, 1-byte binary year data	PZ	2-8 byte packed decimal with sign and first digit ignored	PSI	1-8 byte packed decimal with sign ignored	ZSI	1-8 byte zoned decimal with sign ignored
Code	Data Alignment On																												
H	Halfword boundary																												
F	Fullword boundary																												
D	Doubleword boundary																												
Format	Data Size/Type																												
Y2C	2-digit, 2-byte character year data																												
Y2Z	2-digit, 2-byte zoned decimal year data																												
Y2S	2-digit, 2-byte character year or zoned decimal year data. Used when the data contains header and or trailer records.																												
Y2P	2-digit, 2-byte packed decimal year data																												
Y2D	2-digit, 1-byte decimal year data																												
Y2B	2-digit, 1-byte binary year data																												
PZ	2-8 byte packed decimal with sign and first digit ignored																												
PSI	1-8 byte packed decimal with sign ignored																												
ZSI	1-8 byte zoned decimal with sign ignored																												
v	May be specified for variable-length records only to indicate the remainder of the variable-length record, beginning at position v, is to be included in the reformatted record. The value for v must be beyond the minimum record length.																												

Field	Description
f	<p>Specifies a filler field to be inserted into the reformatted input record. Select one of the following formats:</p> <p>Format Filler</p> <p>nX n blank (x '40') bytes</p> <p>nZ n binary zeros (x '00') bytes</p> <p>nH n high-values (x 'FF') bytes</p> <p>nC'...' the specified character string repeated n times</p> <p>nX'...' the specified hexadecimal string repeated n times. The hexadecimal string must be of an even length, representing a whole number of bytes.</p> <p>n may not exceed 32767. For variable length records, f must not be the first field.</p>

2.5.11.1 INREC/OUTREC Control Statement Notes

- The INREC and/or OUTREC statements may not be used when ADDROUT has been specified on the OPTION control statement.
- The effective output record length (^l2 for INREC, ^l3 for OUTREC) will be calculated by the package and need not be defined on the RECORD statement. Messages CASD075I/CASD100I display the length value of records reaching the SORTOUT file (after INREC/OUTREC). Correct specification of the BLKSIZE parameter on the OUTFIL control statement is necessary.
- Fields in the INREC or OUTREC statements may overlay and be sort/merge control fields.
- For variable-length records, the first entry in the FIELDS= parameter must specify or include the four-byte record descriptor word (RDW).
- Variable-length input records will produce variable-length reformatted output records, even if each is of the same length.
- Unicenter CA-Sort will calculate and insert the correct length into the RDW for each variable-length record.
- Since INREC processing occurs after E15 exit processing during Phase 1, a typical use of INREC would be to remove any extraneous data from the records being sorted. This reduces the size of the record actually entering the sort process and will result in less disk work space being used. The OUTREC statement can be used in the same step to reformat the record, after sorting, into its original size and data locations for input to later application programs.
- INREC processing will occur on all records inserted by an E15 exit, and OUTREC processing will occur prior to entering an E35 exit. Therefore, the INREC control fields should reference positions in the input record as they will exist after any processing or alteration by an E15 exit. The OUTREC control fields should reference positions in the record after INREC processing, and before E35 exit processing.
- The INREC control statement may not be used during a merge-only execution. The OUTREC statement may be used.
- The OUTREC control statement may be used to transform date fields into character format. Y2C, Y2Z, Y2S, Y2P, Y2B and Y2D may be used to transform two-digit character, zoned decimal, packed decimal, or decimal year data into four-digit character year data according to the century window that is set. For example, 96 would be transformed to 1996, and 00 would be transformed to 2000. In addition, PZ, PSI, and ZSI may be used with the OUTREC control statement to transform packed decimal and zoned decimal fields, such as month and day, to character fields.

Y2C or Y2Z

May be used to transform two-digit, two-byte character or zoned decimal year data. Both Y2C and Y2Z can be specified in the OUTREC statement. They are interchangeable and can be used for any two-digit field that can be represented as *xyxy*, where *y* is hexadecimal 0-9 and represents a year digit, and *x* is hexadecimal 0-F and is ignored.

For example, 96, represented as X'F9F6', X'F9C6', and X'0906', can be handled by Y2C or Y2Z. Thus, you can use Y2C and Y2Z for two-digit years in many types of dates such as C'yy', C'mm/dd/yy', C'yy.mm.dd', Z'ddmmyy', Z'yyddd', C'dddy' and C'yy-mm'.

Y2S

You must use Y2S *instead of Y2C or Y2Z* on the OUTREC to transform year data for character or zoned decimal 2-digit, 2-byte year data if the data contains header and or trailer records in the year field. Header records contain binary zeroes X'0000' or blanks X'4040' in the year field. Trailer records contain X'FFFF' in the year field.

For example, year 96, represented as X'F9F6' or X'F9C6', would both be transformed to C'1996'. While year 02, represented as X'F0F2' or X'F0C2' would both be transformed to C'2002'. Header and trailer records would be transformed to X'00000000', X'40404040' or X'FFFFFFF'.

Y2P

May be used to transform two-digit, two-byte packed decimal year data. Y2P can be specified in the OUTREC statement and can be used for any two-digit year field that can be represented as *xyyx*, where *y* is hexadecimal 0-9 and represents a year digit, and *x* is hexadecimal 0-F and is ignored.

For example, 96, represented as X'096C', X'096F', and X'8962', can be handled by the Y2P format. Thus, you can use Y2P for two-digit years in many types of dates such as P'yy', P'mmddyy', P'yyymmdd', P'ddmmyy', P'dddy' and P'mmyy'.

Y2D

May be used to transform two-digit, one-byte decimal year data. Y2D can be specified in the OUTREC statement. It can be used for any two-digit year field that can be represented as *yy*, where *y* is hexadecimal and 0-9 represents a year digit.

For example, 96, represented as X'96' can be handled by the Y2D format. Thus, you can use Y2D for two-digit years in many types of dates such as X'yy' and P'yyddd'.

Y2B

May be used to transform two-digit, one-byte binary year data. Y2B can be specified in OUTREC statements. It can be used for any two-digit year field that can be represented by *yy* where *y* is a hexadecimal 0-F.

For example, year 96, represented as X'60' the binary equivalent for decimal 96 can be handled by the Y2B format, and would be transformed to C'1996'.

PZ May be used to transform any two- to eight-byte packed decimal field you want to treat as *xddd...ds*, where *x* is hexadecimal 0-F and is ignored, *d* is hexadecimal 0-9 and represents a decimal digit, and *s* is hexadecimal 0-F and is ignored.

For example, 12, represented as X'012C', X'012D', X'912C', and X'3127', can be handled by PZ, whereas with the PD format, these values would be treated as +12, -12, +912 and -312, respectively. Therefore, you can use PZ for parts of PD fields such as mm, dd, mmdd and ddmm in 'Pmmddy', 'Pddmmy', 'Pyymmdd' and 'Pyymm'.

PSI May be used to transform a packed decimal with sign ignored. PSI can be used for any one- to eight-byte packed decimal field you want to treat as *dd...ds*, where *d* is hexadecimal 0-9 and represents a decimal digit and *s* is hexadecimal 0-F and is ignored.

For example, 123, represented as X'123C', X'123D', X'123F', and X'1231' can be handled by the PSI format, whereas with the PD format, these values would be treated as +123, -123, +123 and -123, respectively. Thus, you can use PSI for parts of PD fields such as ddd in 'Pyydd' and 'Pddy'.

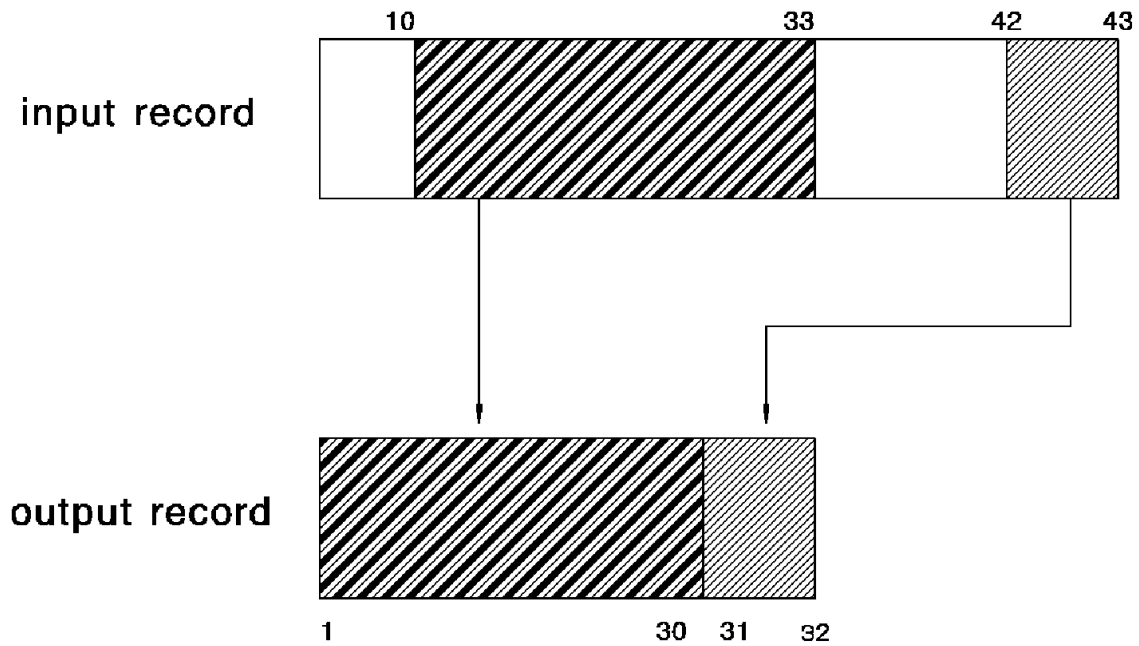
ZSI May be used to transform a zoned decimal with sign ignored. ZSI can be used for any one- to eight-byte zoned decimal field you want to treat as *zd...sd*, where *z* is hexadecimal 0-F and is ignored, *d* is hexadecimal 0-9 and represents a decimal digit and *s* is hexadecimal 0-F and is ignored.

For example, 12, represented as X'F1F2', X'C1D2', X'51C2', and X'0112' can be handled by the ZSI format, whereas with the ZD format, these values would be treated as +12, -12, +12 and -12, respectively. Thus, you can use ZSI for parts of ZD fields such as dd in 'Zyymmdd', ddd in 'Zydd' and mm in 'Zyyddmm' and 'Zyymm'.

INREC/OUTREC Control Statement Examples

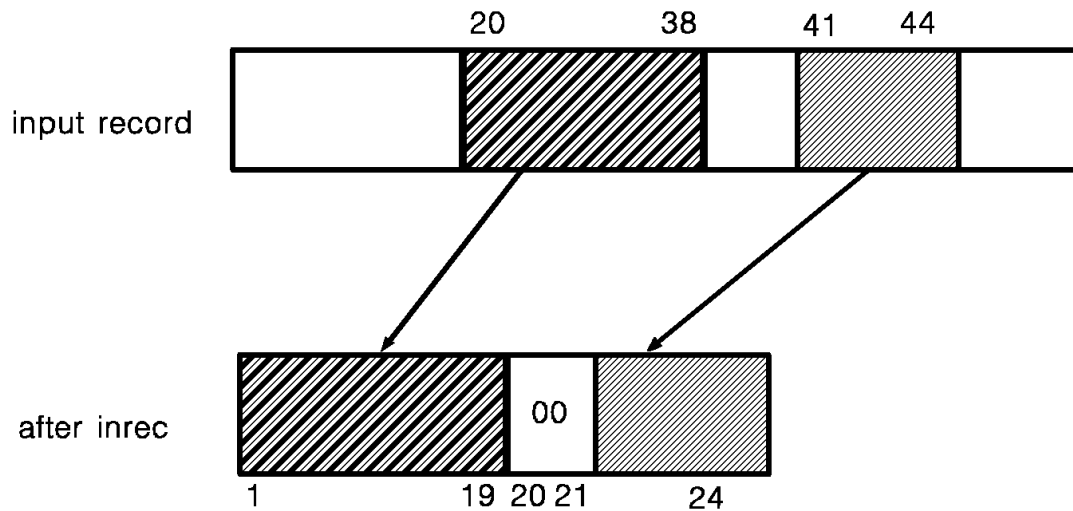
OUTREC FIELDS=(10,30,42,2)

The output record being passed to SORTOUT will be 32 bytes in length and contain bytes 10 through 39 and 42 through 43 of the input record (or record after INREC).



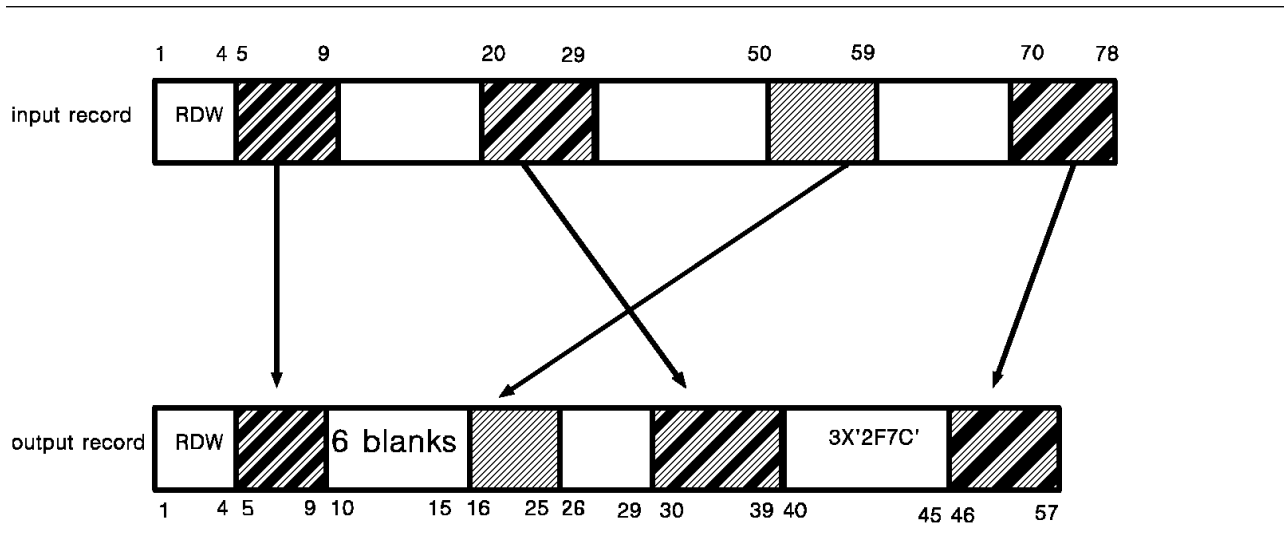
INREC FIELDS=(20,19,41,4,F)

The record entering the sort process will be 24 bytes in length (23 bytes from the INREC statement plus a slack byte due to the alignment parameter), and be composed of positions 20 through 38 and 41 through 44 of the input record. The four bytes selected in the second parameter will actually be moved to byte 20 of the output record to place them on a fullword boundary.



OUTREC FIELDS=(1,9,6X,50,10,30:20,10,3X'2F7C',70)

We assume variable length input records. The first field includes the four RDW bytes plus five bytes of data. The output record will then be padded with six blanks, followed by positions 50 through 59 of the input record. Following that field will be four blanks positioning the next field at location 30. The next field holds positions 20 through 29 of the input record. At the end of each output record will be placed the constant X'2F7C2F7C2F7C' followed by the tail of the variable record starting at position 70 of the input record. Unicenter CA-Sort updates the length of the output record in the RDW bytes automatically. In this case the length is 54 bytes.



The following is a fixed length 13 byte record that contains the following data:

bytes 1,2 two-byte character year data

bytes 3,4 two-byte zoned decimal year data

bytes 5,6 two-byte packed decimal year data

bytes 7-9 three-byte packed decimal Julian date YYDDDS. Note that the year occupies byte 8, and bytes 8 and 9 contain DDD plus a sign 'c'.

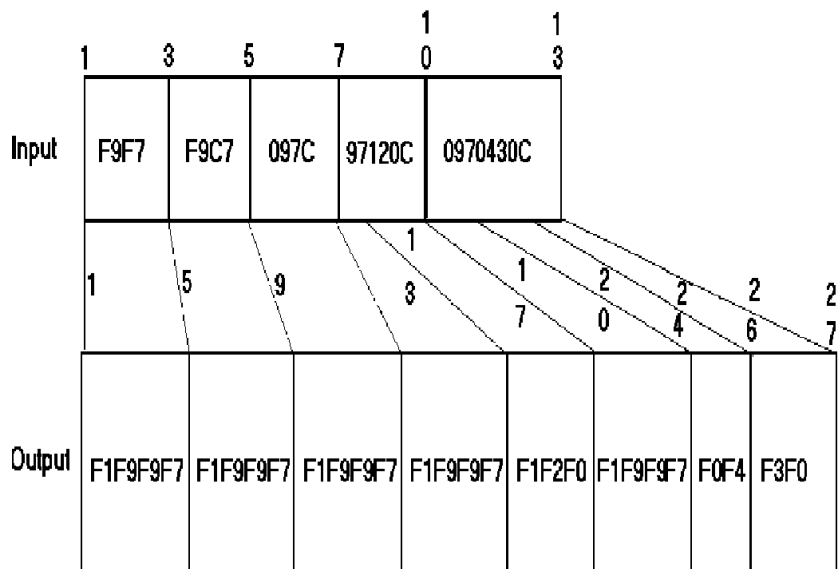
bytes 10-13

four-byte packed decimal date 0YYMMDDS. Note that the year, month, and day are **not** on the byte boundary.

The statement

```
OUTREC FIELDS=(1,2,Y2C,3,2,Y2Z,5,2,Y2P,7,1,Y2D,8,2,PD,10,2,Y2P,
               11,2,PZ,12,2,PZ)
```

will transform the input record into a 27-byte record as shown below.



2.5.12 SUM Control Statement

Allows the summarization or elimination of records containing equal control fields, thus reducing the number of records written to the output.

```
SUM      { FIELDS=(f1,...,fn)                }
          { each of f1,...,fn is p,l,r }
          { FIELDS=(g1,...,gn),FORMAT=rr    }
          { each of g1,...,gn is p,l    }
          { FIELDS=NONE                    }
```

Field	Description
SUM	Statement definer = SUM (used to summarize data fields whenever two records with equal control fields are found).
FIELDS=(Keyword defining the data fields (containing numeric values) which should be summarized.
p	Control-field position relative to the beginning of the logical record. The first byte of a fixed-length record has a relative position of 1. Variable-length data begins in position 5. The maximum value of this parameter is 4092.
l	Length of the summary field in bytes.
r	Data field format (shown following)

f1-n	Length	Signed	Data Format
BI	2 or 4 or 8	NO	Unsigned binary
FI	2 or 4 or 8	YES	Signed binary
ZD	1 - 16	YES	Zoned decimal
PD	1 - 16	YES	Packed decimal

Field	Description
FORMAT=rr	If all data fields have the same format, the r values may be omitted and FORMAT=rr used. The rr value must be specified as BI, FI, ZD or PD.
FIELDS=NONE	Specifies that duplicate equally keyed records should be dropped (deleted). Thus, when two records with the same sort fields are encountered, one of them will be dropped.

2.5.12.1 SUM Control Statement Notes

- Summary fields must not overlap sort/merge control fields or any other summary fields.
- The selection of records to receive the resulting sum and/or records to be deleted is under the control of Unicenter CA-Sort. It cannot be predicted by the user. If specific records must be kept or deleted, the user must perform the summarization using an E15 or E35 exit routine instead of the SUM control statement.
- If overflow occurs during the summarization, the records are left unsummarized.
- Invalid zoned or packed decimal data will cause a data exception and abnormal program termination. General registers 1 and 2 will point to the field being summed into and the invalid data being summed.
- The SUM statement may not be used if the ADDROUT option has been specified on the OPTION control statement, or if SORTx is used.

SUM Statement Examples

SUM FIELDS=(3,4,7,3,20,10),FORMAT=PD

The packed decimal fields, consisting of bytes 3 through 6, 7 through 9 and 20 through 29 will be summarized for all records with equal control fields.

SUM FIELDS=(2,4,ZD,8,2,FI)

Two fields are to be summarized: a zoned decimal field in positions 2 through 5 and a signed binary halfword in positions 8 and 9.

SUM FIELDS=NONE

Duplicate records will be dropped.

2.5.13 ALTSEQ Control Statement

This statement is required when the collating sequence is to be altered in the sort:

```
ALTSEQ CODE=(fftt1 [ ,fftt2 ... ,fftt256 ])
```

Field	Description
ALTSEQ	Statement definier = ALTSEQ (Alternate collating sequence).
CODE=(Code translation definition keyword.
ff	Two hexadecimal digits representing the character whose position in the collating sequence is to be changed.
tt	Two hexadecimal digits representing the new position in the collating sequence the character is to occupy.

2.5.13.1 ALTSEQ Control Statement Notes

- Every sequence definition must consist of four hexadecimal digits in the range of 0-9 or A-F.
- Only control fields which have been defined using the AQ format code will be affected by the new collating sequence. The sequence change is performed on extracted record data, leaving the actual data fields in the record unchanged.
- If no ALTSEQ statement has been defined, then all AQ control fields will be treated as CH fields and collate according to the standard EBCDIC sequence.
- A character may be moved in the sequence only once.
- The moved character (ff) is considered equal to the character already occupying the position to which the character is being moved (tt).
- The character in the tt position is not considered to be moved, however, and it may have its sequence altered as an ff argument, if desired.

ALTSEQ Statement Examples

ALTSEQ CODE=(40FA)

It is sometimes useful to sort fields containing blanks (X'40') after numeric data, rather than before, as in the normal case. The above ALTSEQ statement will translate blanks internally to become the hex code X'FA'. When the records are sorted, the translated areas will sort after the numeric digit, 9 (X'F9').

ALTSEQ CODE=(5BEA,7BEB,7CEC)

Three additional characters exist in the Swedish alphabet that are often represented as the special characters: \$, # and @. However, without sequence modification, these characters would all sort in front of the normal alphabet, rather than after the letter Z. The \$ will be translated to an X'EA' (which will sort after a Z which is, in hex, X'E9'); the # translated to an X'EB' (to follow the \$), and, finally, the @ symbol to an X'EC' (to follow the #).

ALTSEQ CODE=(00F0,01F1,02F2,03F3,04F4,05F5,06F6,07F7,08F8,09F9)

The above ALTSEQ table will cause hexadecimal values X'00'-X'09' to be sorted as if they contained valid zoned decimal numeric data X'F0'-X'F9'. This method will allow zoned decimal and binary data to be mixed, yet both types of values will sort in the same sequence after alphanumeric data.

2.5.14 OPTION Control Statement

This statement provides overrides of default installation options and allows various additional options to be specified.

```
OPTION [ADDROUT={A|D}]  [,CASCADE]  [, {CHALT|NOCHALT}]
      [,CHECK={YES|NO}]  [, {DIAG|NODIAG}]
      [,DSPSIZE={n|MAX}]
      [, {DUMP|NODUMP}]  [,DYNAM={Y|N}]  [,DYNAMFI={Y|N}]
      [, {EQUALS|NOEQUALS}]  [,FILNM=(output,input1,...input9,work)]
      [,KEYLEN=n]  [,LABEL=(output,input1,...input9)]
      [,NOINC]  [,NOMIX]  [,NOTPMK]  [,NRECS=n ]
      [,OPT={D|E}]  [, {PARAMS|PARAMS=name}]  [,PREOPEN={YES|NO}]
      [,PRINT={ALL|NONE|CRITICAL}]  [, {RC|NORC}]
      [,ROUTE=xxx]  [,SKIPREC=n]
      [,SORTIN=(n1,...n9)]  [,SORTOUT=n]
      [,SORTWK=(n1,...n9)]
      [,STORAGE={n|nK|(nK,VIRT)}]  [,SYSNO=n]  [,TP]
      [, {VERIFY|NOVERIFY}]  [,WKASN={ON|OFF}]  [,WORKNM=xxxx]
      [,Y2PAST={F|S}]
```

Field	Description
OPTION	Statement definer=OPTION (sort/merge options)
ADDROUT=x ADDROUT	<p>Specifies that only the direct access addresses of the input records and, optionally, the control fields are to be sorted for a sequential disk or VSAM file. The value x may be:</p> <p>A The output consists only of the direct access address of the input records. This is the default if not specified.</p> <p>D The output consists of the direct access addresses and control from the input records. Not allowed with VSAM input.</p> <p>SAM files: Addresses for these files are ten-byte binary numbers in the form:</p> <p style="text-align: center;">MBBCCHHRRDD</p> <p>where</p> <p>M File number (SORTIN1=0, SORTIN2=1, and so forth)</p> <p>BB Bin number (always 00)</p> <p>CC Cylinder number, relative to zero</p> <p>HH Head number (track), relative to zero</p> <p>RR Record number (block) on the input file track, relative to zero</p> <p>DD Zero (00) for unblocked fixed-length records, or the displacement, relative to zero, of the record within the block</p> <p>DD Four (04) for unblocked variable-length records, or the displacement within the block</p> <p>VSAM files: Addresses for these files are five-byte binary numbers in the form:</p> <p style="text-align: center;">MAAAA</p> <p>where</p> <p>M File number (SORTIN1=1, SORTIN2=2, and so forth)</p> <p>AAAA Relative byte address for key-sequenced data sets (KSDS) and entry-sequenced data sets (ESDS), else the relative record number for relative record data sets (RRDS)</p>

CASCADE	Indicates the SORTx feature is to use only two work files, SORT1 and SORT2, and perform using a cascading SRAM technique in order to save work space and utilize storage better for large sorts.
CHALT	Specifies that the CH format control fields will be affected by an alternate collating sequence (AQ format control fields).
CHECK=x	Indicates whether record counts should be checked at the end of Phase 3 when an E35 exit is in use. Y Record counts should be checked, and if found to be unequal, message CASD207W will follow. Y (yes) is the default. N Unequal record counts will be accepted without error.
DIAG	Specifies that performance and tuning characteristics are to be issued on the printer.
DSPSIZE	Specifies the size, in megabytes, of the data space being created. If n is zero, no data space is used. If MAX is specified, the size of the created data space will be taken from the DSIZE of the SYSDEF job control statement. The value of this parameter overwrites the DSPSIZE parameter of the CASORT macro. The value of n must not exceed 2048.
DUMP	Errors detected in Phase 0 are to produce a partition dump (default is DUMP after Phase 0). If a run time error occurs, sort will produce a partition dump including partition GETVIS plus any necessary control blocks.
DYNAM=x	Specifies whether the interface to Unicenter CA-Dynam/D is to be used or not. The default value for x is set at Unicenter CA-Sort installation time. Y Use interface. Unicenter CA-Dynam/D must be active. N Do not use interface.
DYNAMFI=x	Specifies whether the interface to Unicenter CA-Dynam/FI is to be used. The default value is set at Unicenter CA-Sort installation time. Y Use interface. Unicenter CA-Dynam/FI must be installed and active. N Unicenter CA-Dynam/FI is not installed and active, or the interface is not to be used.

EQUALS	Records having identical control fields will be written to the output file in the same sequence as they were read. The EQUALS option may cause some performance degradation. EQUALS is not allowed when using SORTx statements. The default value for this parameter is set at installation time.
ERASE	Has no effect. Supported for compatibility.
FILNM=	The default file names (SORTOUT,SORTIN1-9 and the first four characters of SORTWKn) may be overridden using the FILNM parameter.
KEYLEN=n	Specifies that the input file is an unblocked direct access (DA) file with fixed-length records that are preceded by a key field of n bytes. Both input and output must be direct access (DA) files.
LABEL=OUTPUT, INPUT ,.	<p>Indicates the type of label to be associated with each input and output file. The values specified are positional, in the following order: SORTOUT, SORTIN1, SORTIN2, and so forth. The valid codes are</p> <p>S File contains standard VSE labels. This is the default for each file.</p> <p>U File is unlabeled and on tape.</p> <p>N File contains nonstandard and/or additional user header or trailer labels. User exit routines must be provided to process the labels. This could be done in E11, E17, E21, E27, E31 or E37 exit routines.</p>
NOCI	Has no effect. Supported for compatibility.
NODUMP	No partition dump is to be produced in case of an error during any phase of the run.
NOEQUALS	Overrides the EQUALS option and does not preserve the input sequence of equally keyed records.
NOINC	Forces the SORTWK1 work file to be opened and used even if the input could be sorted entirely in-core. Use of this option will degrade the performance of small sorts.
NOMIX	Uses work assignment when calculating work block size instead of values in DISK= at installation. May yield better block size and perform less I/O. This option is equivalent to specifying WKASN=ON during installation or on the OPTION statement.
NORC	Return code will not be set for use with conditional JCL on VSE 2.1 or above. Unicenter CA-Sort will CANCEL if errors occur.

NOTPMK	Specifies that no tape mark is to be written before the first data block on each volume of unlabeled tape output.
NOVERIFY	The default; output blocks will not be reread to ensure they were written correctly.
NRECS=n	Specifies that end-of-file is to be simulated after processing n records from the input files. This option is useful for auditing and test data generation to limit the number of records selected, especially when used with the file copy feature of the MERGE function.
OPT=x	<p>Optimization and work file usage:</p> <p>E Open work file extents sequentially and optimize based on amount of work file I/O. Work file extents will not be opened unless needed and the previous extent has filled. This is the recommended value.</p> <p>D Open all work file extents initially and optimize based on CPU and elapsed time. Secondary allocations will not occur until all extents have been filled.</p>
PARAMS PARAMS=name	Specifies that the user routine SORTPARM is to be called after the control statements have been read to perform specialized overrides of Unicenter CA-Sort execution parameters and options. The user routine name may be changed by specifying the name subparameter; this allows different types of user routines to be coded. Refer to 'Unicenter CA-Sort Extended Features' for complete details on coding user parameter override routines.
PREOPEN=Y/N	Specifies that Unicenter CA-Sort is to open all disk input files in Phase 0 to check the device type. Only necessary if FBA devices are being used on the system. The default is set at generation time.
PRINT	<p>Specifies which type of sort/merge messages are to be printed as follows:</p> <p>ALL all messages</p> <p>NONE no messages</p> <p>CRITICAL only E-level messages</p>
RC	<p>Sets return code to be used with conditional JCL on VSE 2.1 or above. Unicenter CA-Sort will not cancel if errors occur. The following return codes are set by CASORT:</p> <p>0 run successful</p> <p>16 errors occurred</p>

ROUTE=xxx	<p>Specifies where Unicenter CA-Sort messages are printed as follows:</p> <p>LST messages on printer</p> <p>LOG messages on console</p> <p>LAL messages on printer and console</p>
SKIPREC=n	Specifies that n records are to be bypassed from the input file(s) before beginning to accept records into the sort/merge process. This option is useful for auditing or test data generation purposes.
SORTIN= (n ¹ ,...n ⁹)	Indicates which logical units to check against DLBL/TLBL labels when determining SORTINn DTFs. Insert the logical unit number in place of n ¹ -n ⁹ . Logical units in extents override for I/O.
SORTOUT=n	Indicates which logical unit to check against DLBL/TLBL labels when determining SORTOUT DTF type. I/O is done using logical unit in extent.
SORTWK= (n ¹ ,...n ⁹)	Indicates which logical units to check against when determining work file devices. The logical units must be assigned to any DASD or unassigned.
STORAGE=n	The value n specifies the maximum number of bytes of main storage that Unicenter CA-Sort is to use during execution.
STORAGE=nK	Specifies the maximum number of K-bytes of main storage Unicenter CA-Sort will be allowed to use during execution. At least 30K must be available. The default value for n is either the SIZE parameter on the EXEC statement, the partition size (when not invoked), or the difference between the Unicenter CA-Sort load address and the end of the partition for invoked sorts. The smallest of any of these values will be chosen and used.
STORAGE=(nK,VIRT)	The subparameter VIRT is used under VSE to suppress page fixing. Usage of VIRT, however, decreases performance and should be avoided.
SYSNO=nnn	Specifies the new default for the SYSnnn starting logical unit number. The default for n is 000. SORTCKP is then SYSnnn, SORTOUT becomes SYSnnn+1, SORTIN1 is SYSnnn+2, and so forth.
TP	Reduces I/O accesses in TP environments.
VERIFY	Specifies that each block of a disk output file will be checked to insure correct writing (ignored for VSAM files). This option will degrade performance.

WKASN=xx	Work file assignments, using either the default logical units or those found in the work file extent statements, are to be used in determining the work devices in use. The work logical units found to be assigned to DASD will replace the devices specified in the DISK= installation parameter if WKASN=ON is specified. The DISK= values will be used if OFF is specified. Correct assignments with WKASN=ON can aid performance.
WORKNM=xxxx	The first four characters of SORTOUT, SORTINn, SORTWKn, SORTn and SORTOTn file names may be changed from the default SORT to another prefix.
Y2PAST=x	<p>Specifies the type of century window (fixed or sliding) to be used when interpreting 2-digit years. A century window spans a 100 year period, such as 1950 to 2049, and determines how 2-digit years are handled. If the century window is set at 1950 to 2049, then a value of 60 would be interpreted as 1960. If the century window is set at 1970 to 2069, a value of 60 would be interpreted as 2060.</p> <p>F Specifies a fixed date at which the century window is to begin. F can be any value between 1000 and 3000. If, for instance, Y2PAST=1997, then the century window would begin in 1997 and span from 1997 to 2096.</p> <p>S Specifies the number of years before the current year that the century window is to begin. S can be any value between 0 and 100. The default is 80. If, for example, the current window is 1997, by default, the century window would begin at 1997 - 80 or 1917. When the current year becomes 1998, the beginning of the century window will change to 1998 - 80 or 1918. In this case, the century window is now 1918 to 2017.</p>

2.5.14.1 Notes on KEYLEN Operand

If the operand KEYLEN=n is specified:

- Unicenter CA-Sort treats the key field as the first n bytes of the record. This must be considered when coding the control field definitions in the FIELDS operand, the record-length values in the LENGTH operand and the block sizes in the BLKSIZE operand of the INPFIL and OUTFIL statements.
- The output must be fixed-length and unblocked. The output device must be direct access.
- A user routine E15 may be supplied to restructure variable-length and/or blocked input to fulfill the requirement of fixed-length unblocked input.

2.5.14.2 Notes on ADDROUT Operand

- The output may be blocked and on any device if the ADDROUT operand is specified (Tag Sort). Note that in this case the output cannot contain a key field.
- The operand ADDROUT=x may not be specified if the input file is read by the user (operand EXIT on the INPFIL statement).
- The operand ADDROUT=x may not be specified for variable spanned records, ISAM files or if control fields contain data on a bit boundary.
- If the output is written on tape and the operand ADDROUT=n is specified, the block size must be 18 bytes or more. Unicenter CA-Sort ensures that the last output block contains at least 18 bytes.
- If the operand ADDROUT=x is specified, the output block size must be in accordance with the figure on the next page.

Format of ADDROUT Records

SAM files: Output Record for ADDROUT=A

MBBCCHHRDD

The LENGTH operand must specify the following values:

l^2 = 10 bytes plus the sum of the control-field lengths
 l^3 = 10 bytes

VSAM files: Output Record for ADDROUT=A

XYYYY

The LENGTH operand must specify the following values:

l^2 = 5 bytes plus the sum of the control-field lengths
 l^3 = 5 bytes

SAM files: Output Record for ADDROUT=D

MBBCCHHRDD Control Fields

The LENGTH operand must specify the following values:

l^2 = 10 bytes plus the sum of the control field lengths
 l^3 = 10 bytes plus the sum of the control field lengths
plus the amount the output record is to be
lengthened or shortened at exit E35.

2.5.15 Overriding Logical Unit Assignments

- Nonstandard logical unit assignments for the sort/merge input, output and work files may be made in using the SORTIN, SORTOUT and SORTWK keywords. Any logical unit numbers specified on the OPTION statement will override the corresponding defaults. The particular logical unit numbers n or n^1 , $n^1 \dots n$ may be omitted in the list. In this case, the internal sort/merge defaults will be assumed.

Example:

```
SORTIN=(7,8),SORTWK=(10)
```

will result in the following assignments:

```
SORTIN1=SYS007  
SORTIN2=SYS008  
SORTIN3=SYS004 (default)  
SORTWK1=SYS010  
SORTWK2=SYS006 (default for FILES=3, WORK=2)
```

- The default or specified logical units will always be checked against the type of label records found during Phase 0. If an incompatibility is found, message CASD021W will be issued and execution will continue. It is assumed that a disk or tape management system will resolve this incompatibility at OPEN time. If a label record is found, the assignment is only used to augment the DTF type already determined (for example, to set a CKD or FBA device type in the DTF). The DEFDTF value specified at Unicenter CA-Sort generation time will only be used if labels and assignments are not found during Phase 0.
- Unicenter CA-Sort will use the logical unit number in the direct access EXTENT statements to perform I/O against the input, output and work files if different from the specified or default values.
- Work logical units, either defaults or specified in the SORTWK parameter or EXTENT, must either be assigned to a valid DASD device, or be unassigned if the work file is SD or DA. This restriction does not apply for VSAM/SAM work files.
- If WKASN=ON has been specified, the work file block size will be optimized based only on the devices actually assigned. This method provides compatibility with the IBM sort/merge program. Proper device type assignment with WKASN=ON can aid performance in a mixed DASD environment, when all devices specified in the DISK= installation parameter are not used for work files in the sort being executed. All device types which will be used MUST be assigned if a disk space management product is in use and could allocate work files on any device. When work files are assigned to an FBA device, Unicenter CA-Sort will use the 3370 device characteristics for the block size calculation if only DISK=3370 is specified for FBA SORTWK. If the specification at generation time is either DISK=3310 or DISK=(3310,3370), Unicenter CA-Sort will use the 3310 device characteristics for the block size calculation.

2.5.15.1 Notes on the DIAG Option

- The DIAG option is useful when attempting to 'fine tune' Unicenter CA-Sort for best performance. Your Computer Associates Technical Representative will gladly assist you in interpreting the DIAG output, which consists of a number of CASD900I messages and several short PDUMPs of Unicenter CA-Sort partition storage.
- DIAG results will vary, based on the main storage available, work area and devices used, input file size and preordering of the input data. Several DIAGs should be obtained using different data from the same file before attempting to influence performance.
- Before reporting any suspected problem to Computer Associates, the job stream in question should ALWAYS be rerun with DIAG specified to provide valuable diagnostic information.
- DIAG may be specified either on the Unicenter CA-Sort OPTION control statement or by specifying DIAG in the SYSPARM field of the VSE JCL OPTION statement as: // OPTION SYSPARM='DIAG'. Place the JCL OPTION statement before the EXEC statement. This method is useful for obtaining information on invoked sorts without the need to recompile the program.

2.5.15.2 OPTION Control Statement Examples

```
OPTION PRINT=CRITICAL,LABEL=(N,S)
```

Suppresses all Unicenter CA-Sort messages except those which indicate a terminal error. It also indicates that the SORTOUT file is a tape volume with nonstandard labels and SORTIN1 has standard labels.

```
OPTION STORAGE=300K,DIAG,ROUTE=LAL
```

Sets a maximum main storage limit of 300K, allow diagnostic and tuning messages to be printed and route all Unicenter CA-Sort messages to the printer and the console.

```
OPTION FILNM=(TOUT,,INP2)
```

Sets the following file names:

```
output = TOUT
input 1 = SORTIN1 (the default)
input 2 = INP2
```

2.5.16 ANALYZE Control Statement

Specifies that Unicenter CA-Sort is to calculate the maximum and minimum work area required on a direct access device, along with a number of other relevant execution characteristics for the associated sort.

ANALYZE CALC

Field	Description
ANALYZE	Statement definer=ANALYZE (work area estimates and execution characteristics).
CALC	Specifies that Unicenter CA-Sort is to calculate the approximate number of tracks or blocks required for sort work area on a direct access device, along with a number of relevant execution characteristics for the associated sorting operation. Unicenter CA-Sort will then cancel at the end of Phase 0 (or return with RC=16 for an invoked sort).

2.5.16.1 ANALYZE Control Statement Notes

- ANALYZE CALC forces the following related options:
DIAG,NODUMP,ROUTE=LST,PRINT=ALL
- The input files are not actually read if ANALYZE CALC is specified; therefore, the SIZE parameter is required on the SORT control statement to estimate the input file size.
- The devices specified in the DISK= parameter of the CASORT option macro will be used when calculating the amount of work space required, unless NOMIX has been specified on the OPTION control statement, and the logical unit for SORTWK1 has been assigned to a DASD device, in which case the estimates will be based upon a work device of that type only.
- If the actual number of input records is not greater than the value shown in the CASD045I G=nnnnn message, an in-core sort will probably be performed, and no work files may be required.
- Work space estimates are easily obtained for any sort merely by placing the ANALYZE CALC statement anywhere within the other Unicenter CA-Sort control statements found in the production JCL, supplying an estimate of records on the SORT control statement, and rerunning the job. Since execution terminates after Phase 0, output files need not be mounted or available. Input files must be assigned and must exist, but will not be read in by Unicenter CA-Sort.
- The ANALYZE control statement replaces the CALCAREA option found in earlier versions of Unicenter CA-Sort and IBM sort/merge products.
- The value in effect for WKASN impacts the calculations performed by the ANALYZE function. If WKASN=OFF has been specified, the estimates will be based upon the lowest capacity CKD device specified in the DISK= installation parameter. FBA estimates will be given only if no CKD devices were specified. If WKASN=ON (for example, NOMIX) is in effect, the estimates will be based only on work file logical units found to be currently assigned to DASD devices. If all work file logical units are found to be unassigned, the values specified for DISK= will be used; otherwise, the DISK= parameter will be ignored.

2.5.17 END Control Statement

Causes Unicenter CA-Sort to stop reading the sort control statements. This must be the last Unicenter CA-Sort statement in the input stream.

END

Field	Description
END	Statement definer=END (delimits the input control statements).
(No operands)	

2.5.17.1 END Control Statement Notes

- Unicenter CA-Sort stops reading the input parameters when the END statement is encountered. The user may then resume read operations from SYSIPT (for example, statement reading) in user exit routines if necessary until the /* end-of-file is reached.
- The END statement must not be continued.
- If no END statement is present, Unicenter CA-Sort stops reading the input parameters when the /* (end-of-file) statement is encountered.
- If SYSIPT is specified on the INPFIL control statement, the statement data to be sorted should be placed immediately after the END statement, followed by a /*.

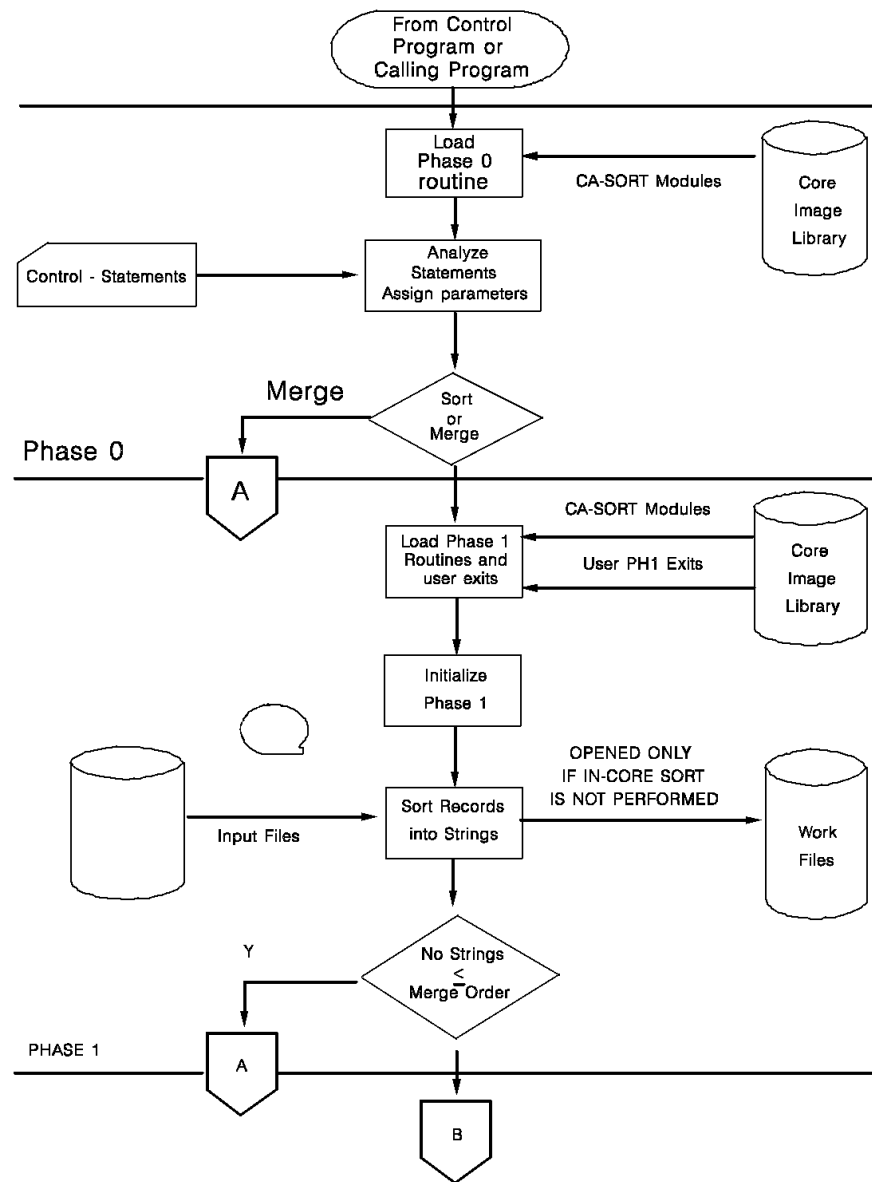
Chapter 3. Defining User Exits

3.1 Overview

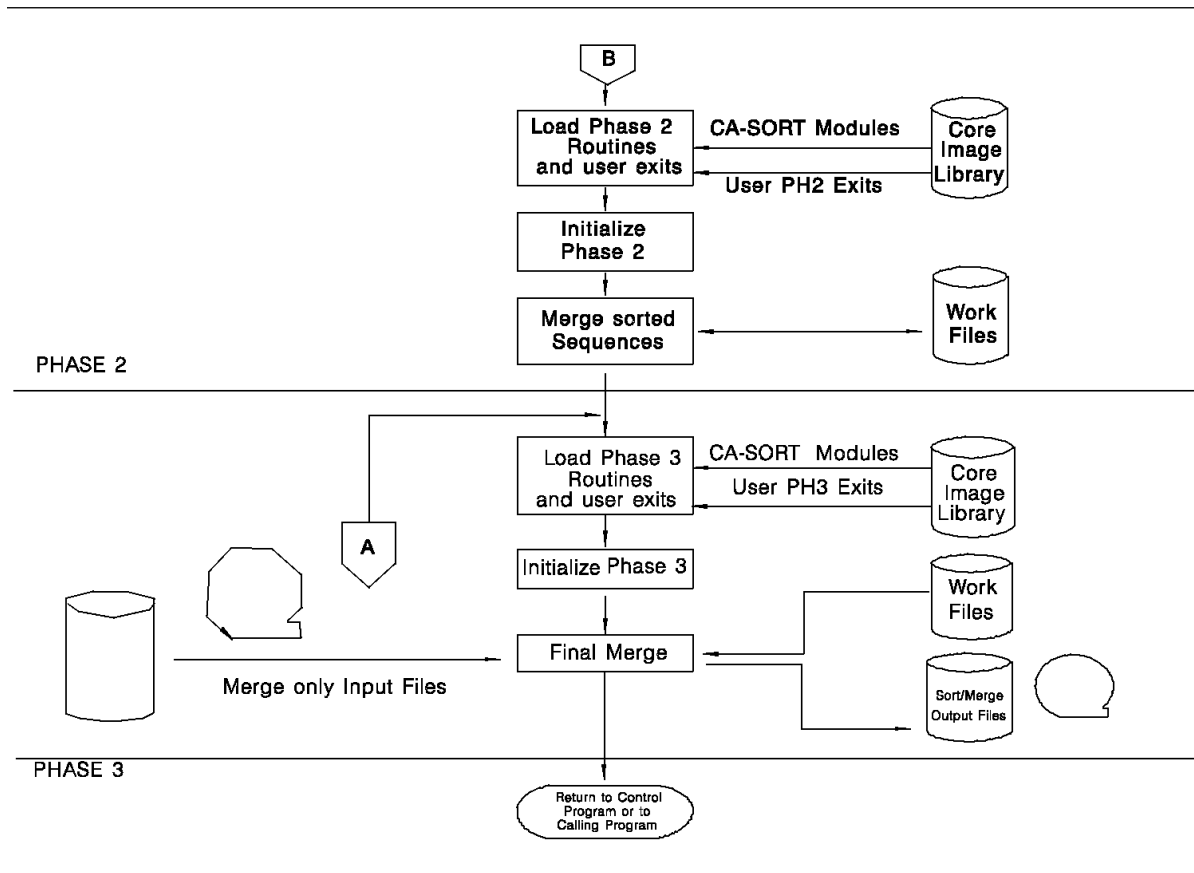
The Unicenter CA-Sort system components (modules) normally perform all the desired sort/merge functions without any external interventions. In some cases, however, it is necessary to modify and to adapt these functions in order to be able to support special processing situations. This can be performed by exit routines (written in the VSE assembler language) invoked at predetermined logical points into the sort/merge system.

As described in the "Introduction" Chapter, the basic structure of the Unicenter CA-Sort system consists of four phases. All exit routines for a particular sort/merge phase are loaded at the start of the respective phase. The functions that may be performed by these routines are dependent upon the current phase of sort/merge execution. This relationship is schematically illustrated in the flowchart on the next two pages.

Loading User Exit Routines



(continued)



User exits may be used in conjunction with record manipulation control statements such as INCLUDE/OMIT, SUM and INREC/OUTREC. The calling sequence for commonly used exits and the record manipulation statements is shown below.

PHASE 1

EXIT E11

EXIT E15

INCLUDE/OMIT,DELBLANK (Sort execution)

INREC

SUM

EXIT E17

PHASE 2

EXIT E21

EXIT E25

SUM

EXIT E27

PHASE 3

EXIT E31

EXIT E32 (Merge execution)

INCLUDE/OMIT,DELBLANK (Merge execution)

SUM

OUTREC

EXIT E35

EXIT E37

3.2 Loading User Exit Routines

When Unicenter CA-Sort is initiated by the EXEC statement, the user exit routines are loaded automatically by the appropriate sort/merge processing phase.

The following rules apply in this case:

- All particular exit routines for a given phase must be treated as a single entity and cataloged under a unique name in the core image library. This name will have to be placed into the appropriate PHn keyword operand in the MODS control statement.
- If the exit routines are relocatable by the system loader, their lengths must be specified in the PHn operand on the MODS statement. Unicenter CA-Sort loads these routines at the most suitable address in the partition.
- If the exit routines are not relocatable, they must be link edited at the highest possible address in the partition where Unicenter CA-Sort is to be run. This address must be located on a doubleword boundary and must be referenced (in its decimal form) in the appropriate PHn operand on the MODS statement.
- If the exit routine is to receive control in AMODE 31, the phase must be link edited with the AMODE 31 attribute.

When Unicenter CA-Sort is initiated by the LOAD macro, one or more exit routines may be preloaded by the user (for example, they are part of the program). The following rules apply in this case:

- The address of the preloaded routine must be placed in the proper position in the Unicenter CA-Sort parameter list.
- The name and loading information must be omitted in the related PHn operand on the MODS statement (by replacing with commas).
- Both preloaded and cataloged exit routines may be used if Unicenter CA-Sort is initiated by the LOAD macro.
- If the exit routine is to receive control in AMODE 31, the high order bit of the routine address must be set to one in the Unicenter CA-Sort parameter list.

Passing Control to User Exits

All of the user exit routines for a given phase are loaded as a single program module. Unicenter CA-Sort will not transfer control to an exit using the entry-point address as resolved by the linkage editor. Instead, control will be transferred to the address in the partition where the exit was loaded. Therefore, the branch table, as described below, must comprise the first executable instructions in the user exit phase. To pass control to a routine at a particular exit, Unicenter CA-Sort uses a branch table, where each four-byte branch instruction (BC type) points to a predetermined exit routine. The branch tables have fixed format for every Unicenter CA-Sort processing phase. Unused positions (exits) in the branch table must be padded by four bytes (for example, A(0) address constant).

The branch tables must be assembled with register 15 as base address. The figure below shows the branch table format for Phases 1, 2 and 3.

Phase 1 Branch Table:		
USING ENTRY1,15		
ENTRY1	B	E11
	B	E15
	B	E17
	B	E18
*PROGRAMMER'S PHASE 1 PROCESSING		
*ROUTINES E11, E15, E17 AND E18		
Phase 2 Branch Table:		
USING ENTRY2,15		
ENTRY2	B	E21
	B	E25
	B	E27
*PROGRAMMER'S PHASE 2 PROCESSING		
*ROUTINES E21, E25, AND E27		
Phase 3 Branch Table:		
USING ENTRY3,15		
ENTRY3	B	E31
	B	E32
	B	E35
	B	E37
	B	E38
	B	E39
*PROGRAMMER'S PHASE 3 PROCESSING		
*ROUTINES E31, E32, E35, E37, E38, E39		

Example

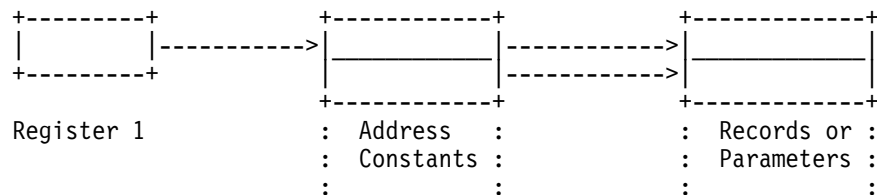
The Branch Table below is for the internal sort phase (Phase 1). It shows that exits E11, E17 and E18 are active and E15 is not used:

PH1	B	E11	-	Branch to E11
	DC	A(0)	-	E15 is not used (padded)
	B	E17	-	Branch to E17
	B	E18	-	Branch to E18

3.3 Parameter Conventions for User Exit Routines

Unicenter CA-Sort makes use of the VSE standard linkage conventions and will place the necessary information into general purpose registers 1, 13, 14 and 15 at each entry to any exit routine as follows:

Register 1 This register contains the address of a list of address constants. Each address constant, in turn, points to information (for example, records, data fields, and so forth) required by the exit routine. The length of the address list and its content depends on the function of the particular exit routine. The figure below illustrates this parameter passing method.:



Register 13 This register contains the starting address of the Unicenter CA-Sort save area. All registers used in the exit routine must be stored in this area according to the VSE system standards (for example, using the SAVE macro instruction) and restored (by the RETURN macro) before control is passed back to Unicenter CA-Sort.:

Register 14 This register holds the return address to the Unicenter CA-Sort system.:

Register 15 This register contains the entry point of the user exit as resolved by the linkage editor. It is the user's responsibility to insure the entry point of the exit points to the branch table. Register 15 may be used as a base register in the user routine.:

Unicenter CA-Sort does not check the validity of any codes or addresses returned by the exit routines. Any violation of the exit calling conventions will cause unpredictable results.

3.4 User Exit Parameters and Functions

The following table describes the Unicenter CA-Sort exit parameters and their functions. The table following that is a summary of the program exits and their uses. The first digit of the exit number is the phase in which the exit is located (1 for the input sort phase, 2 for the intermediate merge phase and 3 for the final merge phase). The second digit represents the type of function that can be performed at an exit. Two types of exits are available in Unicenter CA-Sort. The exits E11, E21 and E31 are assignment component exits; all the other exits are execution component exits.

Table 3-1. Parameters Passed at Sort/Merge Program Exits

Exit(s)	Parameter Number	Function of Parameter
E11,E21,E31	1	Not used (contains zero)
	2	Not used (contains zero)
	3	Address of the number of logical unit containing trailer to be processed
	4	Address of the number of logical unit containing header to be processed
	5	Address of fullword containing input blockcount
	6	Address of logical unit number table
E15	1	Address of new record about to be sorted
	2	Address of action word
	3	Address of record length vector
	4	Address of record type
E25	1	Address of next record to be placed in output buffer
	2	Address of previous record for current sequence (in output buffer)
	3	Address of action word

Exit(s)	Parameter Number	Function of Parameter
E35	1	Address of next record to be placed in output buffer
	2	Address of previous record for current sequence (in output buffer)
	3	Address of action word
	4	Address of sequence check word
	5	Address of record length vector
	6	Address of record type
E17,E37	1	Address of block count fullword
E27		No parameters are passed at this exit
E18,E38	1	Address of CCB for file being read when error occurred
(non-VSAM)	2	Address of channel program being executed when error occurred
	3	Address of action word
E18,E38,E39	1	type indicator
(VSAM)	2	Not used (contains zero)
	3	Address of action word
E39	1	Address of CCB of file being written
	2	Address of channel program being executed when error occurred
E32	1	Address of merge input record
	2	Input file number
	3	Address of action word
	4	Address of record length vector
	5	Address of record type

Uses for Sort/Merge Program Exits

Uses of Exits	Phase 1				Phase 2			Phase 3					
	E11	E15	E17	E18	E21	E25	E27	E31	E32	E35	E37	E38	E39
Assignment	X				X				X				
Label processing	X		X					X			X		
Opening files	X				X			X					
Closing files	X		X				X	X			X		
Reading input files		X											
Counting input records		X											
Deleting records		X				X				X			
Shortening records		X								X			
Modifying data content of records*		X							X	X			
Modifying control fields*		X							X	X			
Summarizing records						X				X			
Substituting records (merge)									X				
Suppressing sequence checking										X			
Writing output file										X			
Processing read errors				X								X	
Processing direct access write errors													X
Providing VSAM passwords				X								X	X
Providing VSAM Exit Lists				X								X	X

* Does not include lengthening or shortening of control fields or data records.

3.5 Phase 1 Exits

Unicenter CA-Sort allows the user to perform certain tasks by four Exits in Phase 1 (input sort phase).

3.5.1 Exit E11

Calling sequence: Register 1 points to a list of six parameters (address constants). These parameters are as follows:

Parameter	Content
1	Binary 0
2	Binary 0
3	Address that points to a halfword which contains the symbolic (logical) unit number to be used for trailer label processing.
4	Address that points to a halfword which contains the symbolic (logical) unit number for header label processing.
5	Address of fullword containing input block count.
6	Address of logical unit number table.

Called at: E11 is called at the start of Phase 1 and/or whenever an input file with nonstandard labels is to be opened or closed (except for the last close).

3.5.1.1 Notes

- The parameters 3 and 4 contain binary 0 if the input file consists of a single volume.
- The parameters 3 and 4 contain binary 0 at the first entry to E11.
- Parameter 3 contains binary 0 if no trailer labels are to be processed.
- Parameter 4 contains binary 0 if no header labels are to be processed.
- If the halfword containing the logical unit number (pointed to by parameter 3 or 4) is placed by the user into the DTF tables used for OPEN, bit 7 of this halfword must be set to '1'.
- The exit E11 must be used to open input files which have nonstandard labels and process these.
- The exit E11 must be used to close all but the last input file which have nonstandard labels (tape files only).
- The logical unit number table has the following format:

byte ---->	1	2	3		10	11	19
	SORTOUT SYSNO	SORTIN1	SORTIN2	...	SORTIN9	SORTWK1	...

Unused numbers contain X'FF'.

3.5.2 Exit E15

Calling sequence:

Register 1 points to a list of four parameters (address constants). These parameters are as follows:

Parameter	Content
1	Address of the logical record to be processed by the internal sort phase (PH1).
2	Address of Action Word, which tells Unicenter CA-Sort what action to take when control is returned to it (fullword return code).
3	Address of record-length vector l ¹ , l ² , l ³ , l ⁴ , l ⁵ (5 fullwords).
4	Address of record type. Record-type switch (1 byte) containing X'80' for fixed-length records and X'40' for variable-length records.

Called at:

E15 is called whenever an input record is about to be processed by Phase 1. The Action Word is a fullword value and is aligned on a fullword boundary. The Return Code, placed in the Action Word by the exit, must be right-justified and zero-padded.

Return Code (Action Word)	To Be Taken by Unicenter CA-Sort
0	Input record is to be processed.
4	Input record is not to be processed (deletion).
8	Do not return to exit E15.
12	Insert a record.
16	Terminate the sort (stop the task by requesting EOJ).
20	Terminate the input (EOF).

3.5.2.1 Notes

- Before E15 returns control to Unicenter CA-Sort with the Action Word=decimal 12 (Insertion), the logical record address of the record to be inserted must be placed in the first fullword parameter.
- When inserting records, the original input record address is restored by Unicenter CA-Sort before reentering exit E15.
- At end-of-file Unicenter CA-Sort preloads the first parameter with binary 0.
- Records may be inserted after end-of-file.
- If the operand EXIT is specified on the INPFIL statement,
 - The first parameter contains binary 0.
 - Only record insertion is allowed (these records may be read by the user from an input file).
 - The Action Word may only be set to decimal 12 (insertion) or to decimal 8 to indicate the end of record insertion.
 - An Action Word setting of zero is invalid if EXIT is specified; however, it will be treated as an insertion request if a record address is specified, or a return code of 8 if a zero address pointer is presented to Unicenter CA-Sort
- If the operand EXIT is **not** specified on the INPFIL statement:
 - E15 can alter input records before they are processed by Phase 1. The record length may be changed depending on the l² value on the RECORD statement. If the record length is changed, the length value in the record (variable-length) must be changed accordingly.
 - E15 must place one of the return codes shown on the following page in the Action Word (pointed to by parameter 2) before returning control to Unicenter CA-Sort
- If the ADDROUT option is specified for sequential files, records to be inserted must have the following format:

MBBCCCHHRDD	data record
-------------	-------------

<----->

10 Bytes

- If the ADDROUT option is specified for VSAM files, records to be inserted must have the following format:

XXXXY	data record
-------	-------------

<----->

5 Bytes

3.5.3 Exit E17

Calling sequence:

Parameter	Content
1	Address of blockcount fullword.

Called at: E17 is called at the end of Phase 1.

E17 must process the nonstandard tape trailer label of the last input volume and close the last input file.

3.5.4 Exit E18 Without VSAM Files

Calling sequence:

Register 1 points to a list of three parameters (address constants). These parameters are as follows:

Parameter	Content
1	Address of the Command Control Block (CCB) of the file being read.
2	Address of the channel program executing when the error occurred.
3	Address of Action Word (fullword return code).

Called at: E18 is called when an error occurs while reading an input file.

3.5.4.1 Notes

- The BYPASS option on the INPFIL statement is ignored if E18 is activated.
- E18 may not be activated if E38 is activated (only E18 or E38 allowed).
- The CCB can be examined to determine the type of I/O error.
- Before returning control to Unicenter CA-Sort, one of the following decimal return codes must be placed in the Action Word (pointed to by parameter 3):

Return Code	To Be Taken by Unicenter CA-Sort
0	Block is to be deleted (skipped).
4	Block is to be processed.
8	Processing is to be stopped and sort/merge job is to be cancelled.

- If E18 returns control with the Action Word = binary 4, the results are unpredictable if a wrong-length record is accepted.
- If a block is to be examined, E18 should provide a dump before returning control with Action Word=decimal 8.

3.6 Phase 2 Exits

Unicenter CA-Sort allows the user to perform certain tasks by three Exits in Phase 2 (intermediate merge phase).

3.6.1 Exit E21

Calling sequence:

Register 1 points to a list of four parameters (address constants). These parameters are as follows:

Parameter	Content
1	Binary 0
2	Binary 0
3	Binary 0
4	Binary 0

Called at:

E21 is called at the start of Phase 2 and performs initialization functions.

3.6.2 Exit E25

Calling Sequence:

Register 1 points to a list of three parameters (address constants). These parameters are as follows:

Parameter	Content
1	Address of next record to be placed in the output buffer.
2	Address of last record placed in the output buffer (previous record in current sequence).
3	Address of Action Word (fullword return code).

Called at: E25 is activated before each record is moved into an output buffer in Phase 2.

3.6.2.1 Notes

- E25 is not activated for the first record in a sequence.
- If E25 changes the collating sequence, exit E35 should use the sequence-checking suppression.
- Before returning control to Unicenter CA-Sort, one of the following decimal return codes must be placed in the Action Word (pointed to by parameter 3):

Return Code	To Be Taken by Unicenter CA-Sort
0	Normal processing is to continue.
4	Next record is not to be placed in the output buffer. Delete the record pointed to by parameter 1.
16	Terminate the sort (stop the task by requesting EOJ).

3.6.3 Exit E27

Calling sequence:

(No parameters.)

Called at: E27 is called at the end of Phase 2 to perform Close functions.

3.7 Phase 3 Exits

Unicenter CA-Sort allows the user to perform certain tasks by six exits in Phase 3 (final merge).

3.7.1 Exit E31

Calling sequence:

Register 1 points to a list of six parameters (address constants). These parameters are as follows:

Parameter	Content
1	Binary 0
2	Binary 0
3	Address that points to a halfword, which contains the symbolic logical unit number to be used for trailer label processing.
4	Address that points to a halfword, which contains the symbolic logical unit number for header label processing.
5	Address of block count fullword.
6	Address of logical unit number table (see E11).

Called at:

E31 is called at the start of Phase 3 and/or whenever an output file or a merge-only file with nonstandard labels is to be opened or closed (except for the last close).

3.7.1.1 Notes

The same notes apply as with exit E11 with the following exceptions:

- E31 must process merge-only input files with nonstandard labels.
- E31 must process output files with nonstandard labels.

3.7.2 Exit E32 Without INPFIL EXIT

Calling sequence:

Register 1 points to one parameter (address constant).

Parameter	Content
1	Address of merge input record.

Called at:

E32 is called whenever a merge-only input record is about to be processed by Phase 3.

3.7.2.1 Notes

- Merge-only input records may be modified in exit E32, but the record length may not be changed.
- A merge-only input record may be substituted by placing the address of the record, which is to replace the input record, in parameter 1. The record length of this new record must be the same as the length of the record being replaced.

3.7.3 Exit E32 With INPFIL EXIT

Calling sequence:

Register 1 points to a list of five parameters (address constants).
These parameters are as follows:

Parameter	Content
1	Address of next input record.
2	Address that points to a fullword containing the input file number. This number is set by Unicenter CA-Sort as follows: hexadecimal 00 = first file hexadecimal 04 = second file and so forth This code can be used as displacement into a branch table.
3	Address of Action Word (fullword return code).
4	Address of record length vector l ¹ , l ² , l ³ , l ⁴ , l ⁵ (5 fullwords).
5	Address of record type. Record-type switch (1 byte) containing X'80' for fixed-length records and X'40' for variable-length records.

E32 must place one of the following binary return codes in the Action Word (pointed to by parameter 3) before returning control to Unicenter CA-Sort

Return Code	To Be Taken by Unicenter CA-Sort
8	End of file.
12	Process the record.
16	Terminate the sort/merge (stop the task by requesting EOJ).

3.7.4 Exit E35

Calling sequence:

Register 1 points to a list of six parameters (address constants). These parameters are as follows:

Parameter	Content
1	Address of the next record to be placed in the output buffer.
2	Address of the last record placed in the output buffer (previous record in the sequence).
3	Address of Action Word (fullword return code).
4	Address of Sequence Check Word (fullword code).
5	Address of record length vector l ¹ , l ² , l ³ , l ⁴ , l ⁵ (5 words).
6	Address of record type. Record type switch (1 byte) containing X'80' for fixed-length records and X'40' for variable-length records.

Called at:

E35 is called before each record is moved into the output buffer in Phase 3 (sort or merge).

3.7.4.1 Notes

- If the operand EXIT is specified on the OUTFIL statement:
 - The second parameter contains binary 0.
 - The Action Word may only be set to binary 4 for record deletion or to binary 8 or 16 to indicate the end-of-sort/merge.
 - If the Action Word contains a value of zero, Unicenter CA-Sort will interpret the invalid return code as a 4, thus deleting the record.
- If the operand EXIT is not specified on the OUTFIL statement, E35 can alter output records before they are placed in the output buffer. The record length may be changed depending on the I³ value on the RECORD statement. If the record length is changed, the length value in the record (variable-length) must be changed accordingly.
- E35 must place one of the following decimal return codes in the Action Word (pointed to by parameter 3) before returning control to Unicenter CA-Sort:

Return Code	To Be Taken by Unicenter CA-Sort
0	Output record is to be placed in output buffer.
4	Output record is not to be placed in output buffer (deletion).
8	Do not return to E35.
12	Insert a record.
16	Terminate the sort/merge (stop the task by requesting EOJ).

- Before E35 returns control to Unicenter CA-Sort with Action Word=decimal 12 (Insertion), the logical record address of the record to be inserted must be placed in the first fullword parameter.
- When inserting records, the original output record address is restored by Unicenter CA-Sort before reentering exit E35.
- If a record is inserted that does not collate properly in the output sequence, a nonzero value must be placed in the parameter 4 fullword (if the control fields have been modified).
- At end-of-file, the first parameter fullword contains binary 0.

3.7.5 Exit E37

Calling sequence:

Register 1 points to a parameter list as follows:

Parameter	Content
1	Address of blockcount fullword.

Called at: E37 is called at the end of Phase 3.

E37 must process the nonstandard tape trailer label of the last output (and/or merge-only) volume and close the last output (and/or merge-only) file.

3.7.6 Exit E38 Without VSAM files

Calling sequence:

Register 1 points to a list of three parameters (address constants). These parameters are as follows:

Parameter	Content
1	Address of the Command Control Block (CCB) of the file being read.
2	Address of the Channel Program executing when the error occurred.
3	Address of Action Word (fullword return code).

Called at: E38 is called when an error occurs while reading a merge-only input file.

The same notes apply as with exit E18.

3.7.7 Exit E39 Without VSAM files

Calling sequence:

Register 1 points to a list of two parameters (address constants).
These parameters are as follows:

Parameter	Content
1	Address of the Command Control Block (CCB) of the disk file being written.
2	Address of the disk Channel Program executing when the error occurred.

Called at: E39 is called when the system's error-recovery procedures have unsuccessfully attempted a write operation on a direct access device.

3.7.7.1 Notes

- When E39 returns control to Unicenter CA-Sort, the sort/merge is terminated.
- As the VSE control program does not return control to Unicenter CA-Sort when a permanent tape write error occurs, E39 cannot be invoked for tape output files.

3.7.8 Exits E18, E38 and E39 with VSAM Files

Exits E18 (sort input), E38 (merge input) and E39 (sort/merge output) may be used with VSAM files to support the following additional functions:

- VSAM passwords
- VSAM exit lists

Exits used will be entered twice: the first time will request the password list and the second will request the exit list. The first parameter passed to each exit will contain a three-character type indicator plus a padding byte set by Unicenter CA-Sort to denote which function is being requested.

The VSAM Password Calling Sequence (E18, E38, E39)

Register 1 points to a list of three parameters:

Parameter	Content
1	Characters C'PWD' justified left.
2	Pointer to the password list (must be provided by the user).
3	Address of Action Word.

The password list (parameter 2) points to:

Number of entries (binary halfword)	File name 8 bytes	Password 8 bytes	File name	Password
	'-----'		'-----'	
	First file entry		Additional entries	

The pointer to the password list must be set in the respective exit routine by the user. If the password is shorter than 8 bytes, it must be padded with spaces.

The return codes passed by the user in the Action Word must be as follows:

Return Code	To Be Taken by Unicenter CA-Sort
0	No password list provided.
4	Password list provided.
8	Password list provided. Do not return to this exit.

The VSAM exit list calling sequence for E18, E38 and E39 is as follows:

Register 1 points to a list of three parameters:

Parameter	Content
1	Characters C'EXL' justified left.
2	Pointer to the exit list (provided by the user).
3	Address of the Action Word.

The exit list pointed to by the second parameter must be formatted as defined in the supervisor and I/O Macros. Any routines pointed to by the list must return to VSAM using standard VSAM linkage conventions. These routines must not use or alter Register 13, as it is in use by VSAM during exit processing. User routines are responsible for providing their own save area.

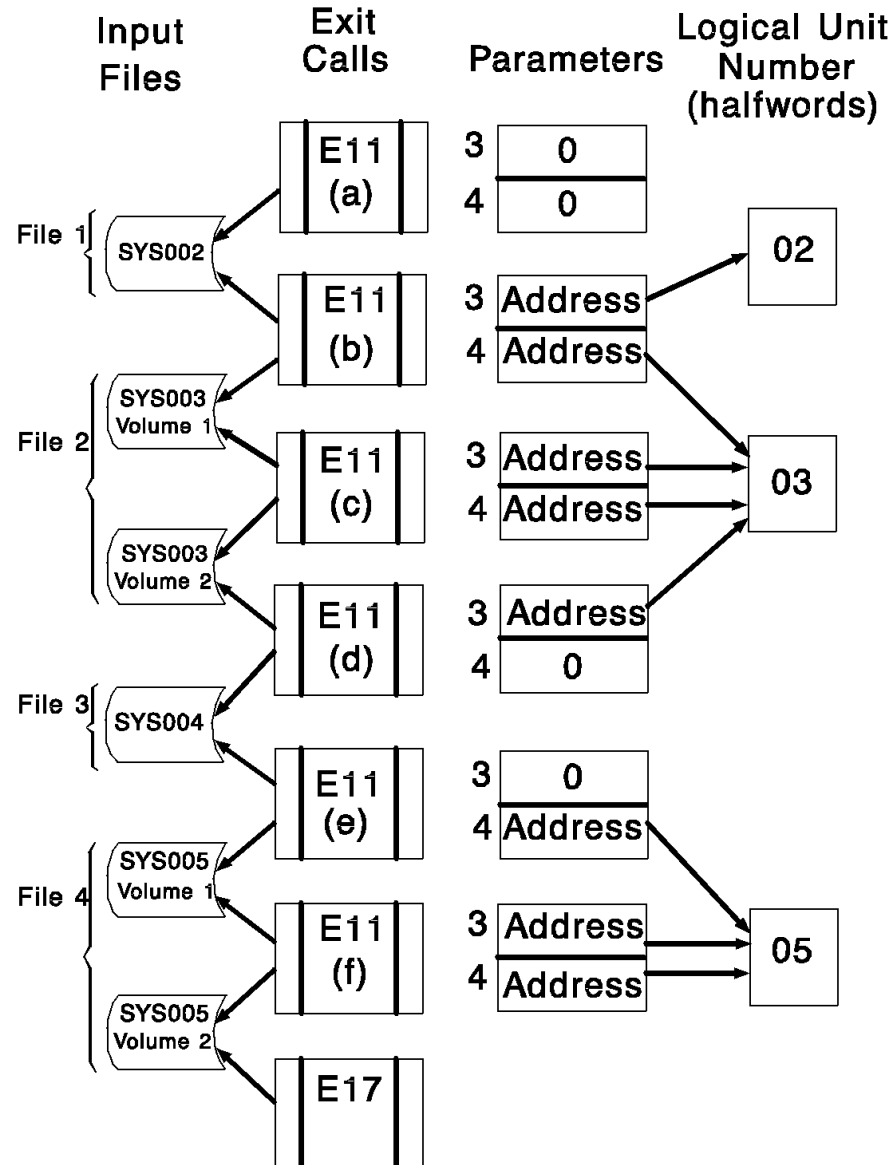
When detecting end-of-file, user routines should provide a LERAD exit and test the feedback code returned (FDBK) for X'04'. Do NOT use the EODAD exit, as Unicenter CA-Sort requires it for its own processing. The feedback code must not be changed in the exit.

When returning to Unicenter CA-Sort from the E18, E38 or E39 exits, one of the following codes must be stored in the Action Word by the user, justified right:

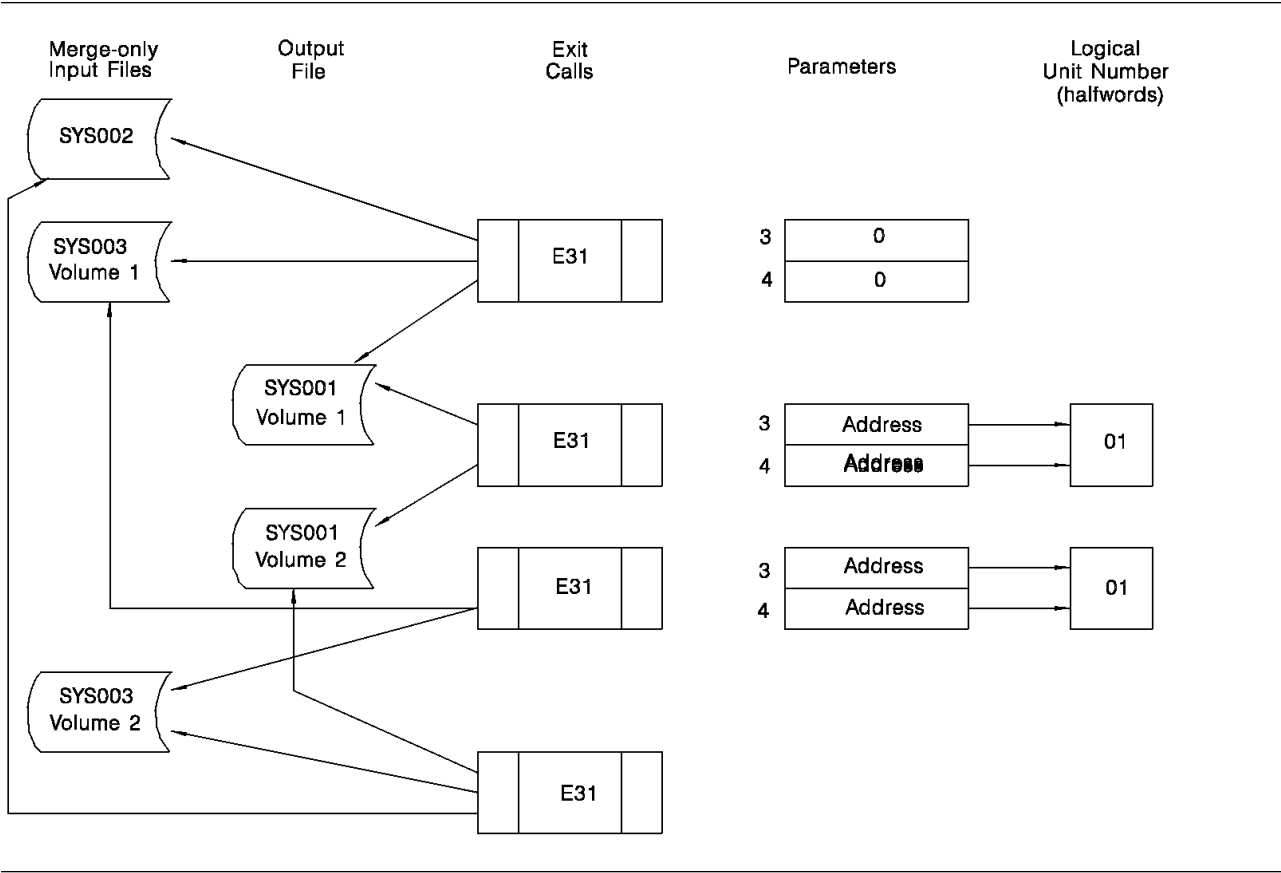
Return Code	To Be Taken by Unicenter CA-Sort
0	No exit list provided.
4	Exit list provided.
8	Exit list provided. Do not return to exit.

3.8 Examples of User Exits

Example 1 A total of four input tape files are to be sorted. Of these files, only file three has standard labels. Files two and four both consist of two volumes. Note that the exit E11 is only activated when nonstandard labels are to be processed.:



Example 2 Two input tape files are to be merged (merge-only). Both input and output files have nonstandard labels. Note that exits E31 and E37 must handle both input and output label processing.:



3.8.1 Sample E15 Exit

```

*****
*                               SAMPLE SORT E15 EXIT                               *
* TWO RECORDS ARE RELEASED TO SORT FOR EACH INPUT CARD READ                       *
*****
*
CAEX15  CSECT
        USING START,15                .ESTABLISH INITIAL BASE REGISTER
START   DC    A(0)                    .EXIT E11 NOT USED
        B      EXITE15                .PROCESS EXIT E15
        DC    A(0)                    .EXIT E17 NOT USED
        DC    A(0)                    .EXIT E18 NOT USED
        DROP   15                    .PREVIOUS BASE REG IS USED BY LIOCS
        USING  START,11                .ESTABLISH NEW BASE REG
EXITE15 EQU    *
        SAVE   (14,12)                .SAVE SORT'S REGISTERS
        LR     11,15                  .LOAD NEW BASE REGISTER
        LA     2,0(,1)                .REG 2 = ADDR OF RECRD ADDR CONSTANT
        LA     3,4(,1)                .REG 3 = ADDR OF ACTION WORD
        LA     4,FILEIN                .REG 4 = ADDR OF INPUT DTF
OPENFILE NOP  GETFILE                .BYPASS OPEN AFTER FIRST TIME
        OPENR  (4)                    .OPEN THE INPUT FILE
        OI     OPENFIL+1,X'F0'        .SET NOP TO BYPASS OPEN
GETFILE NOP   SECOND                  .SWITCH SET TO BYPASS GET WHEN
        GET    (4)                    .READ AN INPUT RECORD
        LA     5,RECONC                .REG 5 = ADDR OF FIRST HALF OF REC
        OI     GETFILE+1,X'F0'        .CLOSE SWITCH FOR 2ND RECORD
        B      STORE                  .STORE 1ST HALF ADDR INTO PARM LIST
*
SECOND  LA     5,RECTWO                .REG 5 = ADDR OF 2ND HALF NOW
        MVI    GETFILE+1,X'00'        .CHANGE SWITCH BACK TO NOP
*
STORE   ST     5,0(,2)                .STORE ADDR OF REC IN PARM LIST
        LA     6,12                  .SET RETURN CODE = INSERT RECORD
RETRN   ST     6,0(,3)                .STORE RETURN CODE IN ACTION WORD
        RETURN (14,12)                .RELOAD REGS AND RETURN TO SORT
*
EX15END LA     6,8                    .SET RETURN CODE = DO NOT RETURN
        CLOSER (4)                    .CLOSE THE INPUT FILE
        B      RETRN                  .STORE RETURN CODE AND EXIT
*
RECDIN  DS     OCL80                  .INPUT I/O AREA
RECONC  DS     CL40                   - FIRST PART OF RECORD
RECTWO  DS     CL40                   - SECOND PART OF RECORD
*
FILEIN  DTFCD  DEVADDR=SYSIPT, IDAREA1=RECDIN, EOFADDR=EX15END,
        MODNAME=E15INP
E15INP  CDMOD
*
        END    CAEX15

```

3.8.2 Sample E35 Exit

```

*****
*                               SAMPLE SORT E35 EXIT                               *
*      REMOVE DUPLICATE RECORDS BASED UPON CLIENT ID                          *
*      THE FIRST RECORD OF A DUPLICATE SET IS RETAINED                          *
*****
*
CAEX35  CSECT
        USING START,15                .ESTABLISH INITIAL BASE REGISTER
*
START   DC    A(0)                    .EXIT E31 NOT USED
        DC    A(0)                    .EXIT E32 NOT USED
        B     EXITE35                 .PROCESS EXIT E35
        DC    A(0)                    .EXIT E37 NOT USED
        DC    A(0)                    .EXIT E38 NOT USED
        DC    A(0)                    .EXIT E39 NOT USED
        DROP  15                      .RELEASE OLD BASE REGISTER
        USING START,11                .ESTABLISH NEW BASE REG
EXITE35 EQU    *
        SAVE  (14,12)                 .SAVE SORT'S REGISTERS
        LR   11,15                    .LOAD NEW BASE REGISTER
        LM   2,6,0(1)                 .LOAD PARAM LIST IN REGS 2-6
*                                     ..REG 2 = ADDR OF SORTED RECORD
*                                     ..REG 4 = ADDR OF ACTION WORD
        LTR  2,2                      .TEST FOR END-OF-FILE
        BZ   EX35END                  ..WILL BE ZERO ADDR WHEN NO MORE
        CLC  0(9,2),CLID              .TEST FOR SAME CLIENT ID
        BNE  NOACT                    .REC IS OK IF NOT THE SAME
        LA   6,4                      .ELSE, SET RETURN CODE = DELETE REC
        B    RETRN                    .AND RETURN TO SORT
*
NOACT   LA   6,0                      .SET RETURN CODE = PROCESS REC
        MVC  CLID,0(2)                .SAVE THE NEW CLIENT ID
*
RETRN   ST   6,0(,4)                 .STORE RETURN CODE IN ACTION WORD
        RETURN (14,12)                .RELOAD REGS AND RETURN TO SORT
*
EX35END LA   6,8                      .SET RETURN CODE = DO NOT RETURN
        B    RETRN                    .STORE RETURN CODE AND EXIT
*
CLID    DC   CL9' '
*
        END    CAEX35

```

Note: Duplicate records may also be deleted using the `FIELDS=NONE` parameter of the `SUM` control statement.

Chapter 4. Operating the Sort/Merge

4.1 Overview

This chapter shows how to initiate execution of the Unicenter CA-Sort program.

The Unicenter CA-Sort program can be initiated by means of control statements in the system input stream, or by receiving control directly from an executing program.

4.2 Initiation by Job Control Statements

The following standard Job Control statements are required to define a sort/merge application to the Job Control program.

- | | |
|--------------|--|
| JOB | Jobname is required, together with any installation-defined accounting data. |
| ASSGN | Assign statements are required only if the I/O devices used in an application have not been previously assigned to the appropriate symbolic names used by the sort/merge program and a disk or tape management product, which has the ability to assign logical units at OPEN time, is not in use. |
| EXEC | Program name is the required operand entry. The program name may be selected by the user during the generation process. The SIZE=nK and REAL subparameters are optional. The sort/merge control statements may also be retrieved from the VSE procedure library. If SIZE is specified on the EXEC statement, Unicenter CA-Sort uses this value if it is different from the default partition size. |

4.2.1 Input File Statements

TLBL or DLBL

If an input file contains standard labels, a DLBL (DASD) or a TLBL (tape) statement must be included for each input file. The file name of either type can range from SORTIN1 to SORTINn, where n is 9 or less, depending on the number of input files to be sorted or merged. The file name of the input file device must also be included. Symbolic unit names must be consecutive and in ascending sequence. Refer to Table 4-1 on page 4-11 for default file names and default symbolic unit names.

EXTENT

One EXTENT statement defines the limits of the direct access area of each input file. If a file is spread over more than one area, an EXTENT statement is required for each area. EXTENT statements must include the symbolic unit name of the device containing the extent.

The first extent of each file must be assigned at the time the file is opened to symbolic units specified in Table 4-1 on page 4-11, or by the SORTIN keyword on the OPTION statement. Further extents of the same file may be assigned to any valid logical unit. Split cylinder extents are allowed.

4.2.2 Output File Statements

TLBL or DLBL

If the output file is to have standard labels, a DLBL (DASD) or a TLBL statement (tape) must be provided. The file name on this statement must be SORTOUT; the symbolic unit name must be SYS001 or the value defined by the SORTOUT keyword on the OPTION statement.

EXTENT

One EXTENT statement defines the limits of each direct access area of the output file. The symbolic name of the output unit should appear on this statement. Split cylinder extents are allowed.

4.2.3 Notes on Input and Output File Statements

- When the output file will reside on a 3310, 3370, or 93xx FBA disk device, the CISIZE parameter should be placed on the SORTOUT DLBL statement. The minimum CISIZE that may be specified is: output block size + 7 rounded up to the next multiple of 512. Performance usually improves if a value larger than the minimum is chosen. However, a very large CISIZE may degrade performance.
- If 7-track tapes are used, please note that:
 - for input, the converter or translator must be ON, depending upon the mode in which the data was created.
 - for output, either the translator or the converter feature must be selected.
- EXTENT statements are not required if a disk space management product is in use which is able to provide them. If the file being defined is a VSAM-managed SAM file under Release 2 of VSE, extents are only required if a work or output data set using implicit definition is to be opened.

4.2.4 Work Area Statements

JCL describing the intermediate work areas need only be provided if the input file is too large to be sorted in-core. Unicenter CA-Sort will only issue an OPEN for the work file if available main storage is exhausted before the sorting process has completed, or if the parameter NOINC has been specified on the OPTION Control Statement.

DLBL If work areas are used, one DLBL statement must be included for each work file. The file name must be SORTWK1 through SORTWK n , where n is up to 9. The symbolic unit names assigned to the work areas must appear on the associated EXTENT statements. See Table 4-1 on page 4-11 for unit name defaults. Nonstandard unit names may be specified by the SORTWK keyword on the OPTION statement. The CISIZE parameter should be omitted when the work files will reside on an FBA DASD device.

EXTENT One EXTENT statement should be included to define each direct access work area. Secondary allocations are supported for all SD sort work files. The total area may be divided into a maximum of 255 extents. An extent on CKD DASD should be at least 3 tracks. An extent on FBA DASD should be at least 150 blocks. No split cylinder extents are allowed.

- When WORK=DA is specified on the SORT statement, the DLBL Job Control statement must show the file type as DA. Only one DLBL statement with file name SORTWK1 may be used if WORK=DA is specified. The associated EXTENT statements must show symbolic unit names (logical unit numbers) in ascending order when extents are allocated on different volumes. The first one must be either the default shown in Table 4-1 on page 4-11 or the value set by the SORTWK keyword.
- Because Unicenter CA-Sort allows up to 50 total allocations on sort work extents, DA-type work files are retained mainly for compatibility reasons.

Example:

```
// DLBL SORTWK1,,1,DA
// EXTENT SYS003,.....
```

4.2.5 Checkpoint File Assignments

The checkpoint file must be resident either on tape or on disk device as a standard labeled file (file name SORTCKP) assigned to logical unit SYS000 or the number defined by the SYSNO parameter. Sort restart is performed using the standard VSE system // RSTRT statement.

4.2.6 Notes on Work File Specifications

- Intermediate work files are not required for every sort run and only need to be specified if it is known that the file cannot be sorted entirely in-core. Otherwise, they may be omitted.
- Any combination of DASD devices (CKD or FBA) which are supported on the user's system may be specified for work files if they are needed.
- When needed, the total space required corresponds to the file size to be sorted plus a reserve of approximately 10 percent. Depending upon the available main storage and mixed devices used, a reserve of up to 25 percent may be required.
- Specification of the WORK=0 parameter on the SORT control statement is accepted, but its use is discouraged. It is recommended that at least one work file be provided with every sort run in the event that a large input file is encountered unexpectedly. Unicenter CA-Sort will only open the extents provided if the file cannot be sorted in-core.
- The value given in message CASD045I G=nnnnn gives the approximate number of records that may be sorted in-core. If the input file is smaller than this value, the work files will probably not be used.

4.2.7 Notes on Work File Secondary Allocations

Unicenter CA-Sort supports secondary allocation for work files as follows:

- Users without a disk space management product may provide additional extents for all SORTWK_n files. Unicenter CA-Sort will open each extent as needed.
- Users with a disk space manager (including Unicenter CA-Dynam/D) may provide secondary allocations for the last SORTWK_n file being used (for example, SORTWK2 if WORK=2). The extents may be provided in the JCL or generated dynamically by the product. This applies to VSAM-managed SAM data sets as well.
- In all cases, when all available extents provided, either explicitly or dynamically, have been exhausted, Unicenter CA-Sort will continue to issue OPENs against the last SORTWK file until the space requirements have been met. This may result in the IBM message:

4450A NO MORE AVAILABLE EXTENTS

The operator may then enter the location of additional space which may be used by Unicenter CA-Sort, or the sort may be canceled from the console. This operator communication eliminates the 'sort capacity exceeded' condition for sequential work files, and the added flexibility reduces the need for reruns.

4.2.8 Notes on Logical Unit Assignments

- In prior releases of Unicenter CA-Sort, as was the custom with the IBM sort/merge products, default logical units were required to be preassigned to the proper device prior to OPEN in order to determine the type of DTF to be built. As of Release 7.0, Unicenter CA-Sort has relaxed or removed many of the restrictions dealing with assignments prior to OPEN, and will use the following hierarchy to determine the type of DTF to be built for the SORTINn and SORTOUT files:
 1. The U subparameter of the LABEL operand on the OPTION control statement will force an unlabeled tape DTF to be built, regardless of the assignments or other label records found in the job stream. This is done because only tape files may be unlabeled. For an unlabeled tape output file, the SORTOUT logical unit must be assigned to a valid tape device to avoid possible I/O errors. In this case, either the default (see Table 4-1 on page 4-11) or the value specified in the SORTOUT keyword of the OPTION control statement will be used.
 2. A DLBL or TLBL found for the associated file will take precedence next.
 3. If a label cannot be found, the assignment of the logical unit will be checked and used if assigned to a disk or tape device.
 4. If neither labels nor an assignment can be resolved, Unicenter CA-Sort will default to the value specified in the DEFDTF operand of the CASORT option generation macro. This value may be TAPE to cause a sequential tape DTF to be built, DISK to cause a CKD disk DTF, FBA to build an FBA DTF, or CANCEL to cause the sort/merge to be terminated in Phase 0 if the DTF determination cannot be made.
- Unicenter CA-Sort will not make an assignment if a logical unit is found to be unassigned or assigned incorrectly. A disk or tape management system may be used to assign the logical unit at OPEN time, however. Failure to resolve the assignment at OPEN will result in the cancellation of the sort/merge with the IBM error message INVALID LOGICAL UNIT, NO FORMAT1 LABEL FOUND, NO LABEL INFORMATION, or other error.
- Users with FBA disk devices should specify PREOPEN=YES in the Unicenter CA-Sort option macro at generation time, to allow Unicenter CA-Sort to open FBA files during Phase 0 to obtain the CISIZE from the VTOC. The SORTOUT logical unit must be assigned in Phase 0 to the proper device type, CKD or FBA. This restriction applies only to SORTOUT, and then only if FBA devices are present on the system and being used in the sort/merge execution. This is not required if Unicenter CA-Dynam/D Release 5.3 or above is active on the system and DYNAMD=Y is specified in the installation Unicenter CA-Sort macro.
- If VSAM-managed SAM files are being used for input, output or work files, the logical unit need not be preassigned. However, Unicenter CA-Sort must be able to locate the DLBL which specifies VSAM in order to properly generate the DTF necessary to use a VSAM/SAM data set.

- Pooling, or overlapping between work files and input or output files, is not allowed by Unicenter CA-Sort. Some early IBM sort/merge programs allowed this type of overlap to occur. Those job streams which utilize this technique will need to be corrected if converting to Unicenter CA-Sort.

Input and output files may overlap each other if a sort is being performed, but never during a merge-only, as all files are open at the same time.

4.2.9 Notes on the Use of VSAM-managed SAM Files

- Using VSE/VSAM under VSE Release 2 and above, input, output and sort work files may be defined as VSAM-managed SAM data sets. Work files and output files may be defined implicitly or explicitly. File definitions for input files must exist in the VSAM catalog prior to use and, therefore, may not be implicitly defined.
- An explicit definition means the file attributes have been placed into the VSAM catalog using IDCAMS prior to use. An implicit definition causes the attributes of a file to be cataloged dynamically at OPEN time by VSAM. The latter method incurs a significant amount of overhead, and sort/merge executions will probably operate slower than if the files were previously defined.
- It is recommended that output files be explicitly defined in the VSAM catalog prior to use to reduce the VSAM OPEN-time overhead. Sort work files are best defined implicitly, since they cannot be processed by any other program but Unicenter CA-Sort. An explicit definition may be used if the RECORDFORMAT operand specifies NOCIFORMAT. Refer to the example later in this section.
- To define a VSAM-managed SAM file implicitly, place the RECSIZE and RECORDS parameters on the DLBL statement describing the file. An EXTENT statement with a volume serial number is required if a model definition is not already present in the VSAM catalog. Any file created from an implicit define will be defined with record format undefined (RECFM=UNDEF). A subsequent program may read a SORTOUT file created in this manner using VSAM/SAM; however, if VSAM is used, a complete block will be presented to the program by the access method and the program must do its own deblocking. To avoid this, explicitly define the file using IDCAMS and specify RECFM=FB or RECFM=VB, as appropriate.
- On input, a mixture of native VSAM (KSDS, RRDS, ESDS, and so forth) and VSAM-managed SAM files is not permitted unless VSAM is used to read all files and the VSAM keyword is placed on the INPFIL statement. However, mixing VSAM/SAM files with other organizations (ISAM, disk or tape SAM, and so forth) is allowed, and the VSAM keyword should NOT be placed on the INPFIL statement. Sort work files and output files may be of any supported organization or device type, regardless of the input organization. For more information, refer to the INPFIL and OUTFIL control statements in the chapter entitled "Using the Control Language."

The following applies to VSAM/SAM sort work files:

- The parameter DISP=(NEW,DELETE) should always be placed on the DLBL for work data sets.
- Secondary allocation is supported on the last sort work file only (for example, on SORTWK3 if WORK=3). Either explicit or automatic secondary allocation may be used, however, the total number of work extents may not exceed 255.
- The use of the 'Single Extent Primary Allocation' format of definition is not recommended because it will defeat the ability of Unicenter CA-Sort to use secondary allocation.
- As with sequential work files, an extent will only be allocated if needed.
- If a CISIZE is placed on the sort work DLBL, it will be ignored. Unicenter CA-Sort does not write to sort work files in control interval format.

Example of IDCAMS Explicit Definition of a VSAM/SAM Sort Work File

```
// JOB IDCAMS MAINTENANCE
* DELETE AND DEFINE CLUSTER FROM MASTER CATALOG
// EXEC IDCAMS,SIZE=100K
DELETE %SORT.WORKFILE CLUSTER PURGE
DEFINE CLUSTER (NAME (%SORT.WORKFILE) -
                RECORDFORMAT (NOCIFORMAT) -
                TRACKS (3000 1500) -
                VOLUMES (VSAMWK) -
                NONINDEXED NOALLOCATION REUSE SPEED) -
                DATA (NAME (%SORT.WORKFILE.DATA))
/*
/&
```

Example of Implicit Definition of a VSAM/SAM Sort Work File

```
// DLBL SORTWK1,'JOB.ABC.SORTWK',,VSAM,RECORDS=5000,RECSIZE=100,
                DISP=(NEW,DELETE)
// EXTENT ,,VSAMWK
```

For more information on how to define and use VSAM-managed SAM files, refer to the IBM manual entitled *Using the VSE/VSAM Space Management for SAM Feature* SC24-5192-0.

Table 4-1. Sort/Merge Device Assignment Defaults

			Symbolic	Unit Names	When
<i>Device Use</i>	<i>Filename</i>	<i>Unicenter CA-Sort reads input and writes output.</i>	<i>User E15 exit will read the input file. See Note 1.</i>	<i>User E35 exit will write the output file. See Note 2.</i>	<i>User exit will read input and write output.</i>
Output	SORTOUT	SYS001	SYS001		
Input	SORTIN1	SYS002		SYS001	
	:	:		:	
	:	:		:	
	SORTIN9	SYS(n+1)		SYS(n)	
Work	SORTWK1	SYS(n+2)	SYS002	SYS(n+1)	SYS001
	:	:	:	:	:
	:	:	:	:	:
	SORTWK9	SYS(n+m+1)	SYS(m+1)	SYS(n+m)	SYS(m)
Checkpoint	SORTCKP	SYS000	SYS000	SYS000	SYS000

where

n = the number of input files, as specified in the FILES parameter of the SORT, MERGE or SORTx control statement.

m =

the number of work files, as specified in the WORK parameter of the SORT control statement.

4.2.9.1 Notes

1. The defaults in this position will also apply when the SYSIPT keyword is specified on the INPFIL control statement and the BLKSIZE keyword is not specified.
2. The defaults in this position will also apply when any of the following keywords are specified on the OUTFIL control statement and BLKSIZE is not specified: SYSLST, PUNCH, PCHLST, or CADART.

4.2.10 Examples

The program name used in these examples is SORT. This program name may be chosen freely at Unicenter CA-Sort installation time.

4.2.10.1 Example 1

Merge with record selection, collating sequence modification and summarization:

```
// JOB      EXAMPLE 1
// ASSGN    SYS001,X'282'                                TAPE OUTPUT
// ASSGN    SYS002,X'280'                                TAPE INPUT
// ASSGN    SYS003,X'161'                                DISK INPUT
// ASSGN    SYS004,X'281'                                TAPE INPUT
// ASSGN    SYS005,X'163'                                DISK INPUT
// DLBL     SORTIN2,'MERGE INPUT 2',0
// EXTENT   SYS003,000002,1,0,20,70
// TLBL     SORTIN3
// DLBL     SORTIN4,'MERGE INPUT 4',0
// EXTENT   SYS005,000004,1,0,60,60
// EXEC     SORT
//          MERGE  FIELDS=(18,4,PD,A,25,9,ZD,D,39,4,AQ,A),ORDER=4
//          RECORD TYPE=F,LENGTH=124,DELBLANK=49
//          INPFIL BLKSIZE=1488
//          OUTFIL BLKSIZE=1488
//          ALTSEQ CODE=(40FA)
//          SUM    FIELDS=(122,6,PD)
//          OPTION LABEL=(U,U)
//          END
/*
/ &
```

4.2.10.2 Example 2

Sort VSAM file (VSE only):

```
// JOB      EXAMPLE 2
// ASSGN    SYS001,3330,VOL=333333,SHR          DISK OUTPUT
// ASSGN    SYS002,3330,VOL=333333,SHR          DISK INPUT
// ASSGN    SYS003,3330,VOL=333333,SHR          SORT WORK
// DLBL     SORTOUT,'SORT.OUTPUT',,VSAM
// EXTENT   SYS001,333333
// DLBL     SORTWK1,'SORT.WORK',0
// EXTENT   SYS003,333333,,,6707,950
// DLBL     SORTIN1,'SORT.INPUT',,VSAM
// EXTENT   SYS002,333333
// EXEC     SORT,SIZE=80K
//          SORT  FIELDS=(6,21,A,7.1,4.3,A,5,1,A),FORMAT=BI
//          RECORD TYPE=F,LENGTH=276
//          INPFIL VSAM,TOL
//          OUTFIL ESDS
//          END
/*
/ &
```

Note: The second entry in the FIELDS parameter is specifying the limits of a bit control field, which need not follow byte boundaries.

4.2.10.3 Example 3

Sort with record selection:

```
// JOB      EXAMPLE 3
// ASSGN    SYS001,X'281'                      TAPE OUTPUT
// ASSGN    SYS002,X'282'                      TAPE INPUT
// ASSGN    SYS003,X'162'                      SORT WORK
// TLBL     SORTOUT
// DLBL     SORTWK1,,1
// EXTENT   SYS003,000003,1,0,300,40
// EXEC     SORT
//          SORT  FIELDS=(5,4,CH,D,20,12,BI,A,50,2,CH,A),SIZE=2000,  *
//          FILES=1,WORK=1
//          RECORD TYPE=V,LENGTH=(160,,,65,100)
//          INPFIL BLKSIZE=1544,CLOSE=UNLD
//          OUTFIL BLKSIZE=1544,CLOSE=RWD
//          INCLUDE COND=(50,2,CH,GE,C'10',AND,60,5,ZD,GT,9000)
//          OPTION PRINT=CRITICAL,OPT=E,LABEL=(,U),ROUTE=LOG,
//          STORAGE=32K
//          END
/*
/ &
```

4.2.10.4 Example 4

Sort with mixed input files, input record reformatting and implicit definition of VSAM-managed SAM files for sort work and output files:

```
// JOB      EXAMPLE 4
// ASSGN    SYS001,X'1C1'                                DISK OUTPUT
// ASSGN    SYS002,X'480'                                TAPE INPUT
// ASSGN    SYS003,X'1C1'                                DISK INPUT
// ASSGN    SYS004,X'1C1'                                SORT WORK
// TLBL     SORTIN1,'TAPE INPUT'
// DLBL     SORTIN2,'DISK FILE INPUT',0,SD
// EXTENT   SYS003,PACK01,1,0,2400,12
// DLBL     SORTOUT,'%SORTED.OUTPUT',,VSAM,RECORDS=1000,RECSIZE=100, *
//          DISP=(NEW,KEEP)
// EXTENT   SYS001,VSAMWK
// DLBL     SORTWK1,'%SORT.WORK1',,VSAM,RECORDS=1000,RECSIZE=135, *
//          DISP=(NEW,DELETE)
// EXTENT   SYS004,VSAMWK
// EXEC     SORT,SIZE=200K
//          SORT   FIELDS=(1,9,CH,A,20,5,PD,D),WORK=1,FILES=2
//          RECORD TYPE=F,LENGTH=(135,,100)
//          INPFIL BLKSIZE=7000
//          OUTFIL BLKSIZE=5000
//          INREC  FIELDS=(1,50,10X,51,40)
//          OPTION EQUALS,PRINT=ALL
//          END
/*
/ &
```

4.2.10.5 Example 5

One input file sorted into two sequences using SORTx, the multiple sorted output file facility. The output files are being directed to two different FBA devices.

```
// JOB      EXAMPLE 5
// ASSGN    SYS002,DISK,VOL=CKD050,SHR          INPUT FILE
// ASSGN    SYS003,DISK,VOL=CKD050,SHR          SORT WORK 1
// ASSGN    SYS004,DISK,COL=CKD050,SHR          SORT WORK 2
// ASSGN    SYS020,FBA,VOL=FBA070,SHR          OUTPUT FILE 1
// ASSGN    SYS021,FBA,VOL=FBA010,SHR          OUTPUT FILE 2
// DLBL     SORTIN1,'DISK FILE INPUT',0,SD      INPUT FILE
// EXTENT    SYS002,CKD050,1,0,300,90
*
// DLBL     SORTOT1,'OUTPUT.FILE.1',0,SD,CISIZE=2048  OUTPUT 1
// EXTENT    SYS020,FBA070,1,0,5890,1800
// DLBL     SORTOT2,'OUTPUT.FILE.2',0,SD,CISIZE=2048  OUTPUT 2
// EXTENT    SYS021,FBA010,1,0,3000,1800
*
// DLBL     SORT1,'SORTX.WORK.FILE.1',0,SD          WORK FILE 1
// EXTENT    SYS003,CKD050,1,0,900,120
// DLBL     SORT2,'SORTX.WORK.FILE.1',0,SD          WORK FILE 2
// EXTENT    SYS004,CKD050,1,0,1020,120
*
// EXEC     SORT,SIZE=200K
           SORT1  FIELDS=(50,12,CH,A)
           SORT2  FIELDS=(1,18,BI,D,20,2,ZD,A)
           RECORD TYPE=F,LENGTH=80
           INPFIL BLKSIZE=4000
           OUTFIL BLKSIZE=4000
           END
/*
/ &
```

4.2.10.6 Example 6

Obtain Sort/Merge keys from Unicenter CA-Dynam Catalog using a symbolic name and sort the file to an explicitly defined VSAM-managed SAM file.

The block size information for the SORTIN1 file is assumed to be in the Unicenter CA-Dynam catalog; therefore, the INPFIL control statement is unnecessary. If the record lengths were also in the catalog, the RECORD statement could be omitted as well.

```
// JOB      EXAMPLE 6
// ASSGN    SYS002,TAPE                               INPUT FILE
// ASSGN    SYS003,DISK,VOL=CKD050,SHR                SORT WORK 1
// TLBL     SORTIN1,'UNSORTED DATA'
// DLBL     SORTOUT,'%SORTED.DATA',,VSAM,DISP=(NEW,KEEP)
// DLBL     SORTWK1,'SORT.WORK.FILE.1',0,SD
// EXTENT   SYS003,CKD050,1,0,900,120
// EXEC     SORT,SIZE=128K
//          SORT   KEY=PAYMAST,WORK=1
//          RECORD TYPE=F,LENGTH=(410)
//          OUTFIL BLKSIZE=8200
//          END
/*
/ &
```

4.2.10.7 Example 7

For auditing purposes, a group of records will be printed from the middle of a disk master file, using SKIPREC to bypass unwanted records, NRECS to limit the amount of output, and SYSLST with various options to format and print the data on the printer.

```
// JOB      EXAMPLE 7
// ASSGN    SYS002,FBA,VOL=FBA070,SHR                INPUT FILE
// DLBL     SORTIN1,'INVENTORY.MASTER.FILE',99/365,SD
// EXTENT   SYS002,FBA070,1,0,9000,14000
// EXEC     SORT,SIZE=128K
//          MERGE  FIELDS=COPY
//          RECORD TYPE=F,LENGTH=(814)
//          INPFIL BLKSIZE=2442
//          OUTFIL SYSLST,DUMP,TITLE='INVENTORY MASTER AUDIT LISTING'
//          OPTION SKIPREC=1500,NRECS=35
//          END
/*
/ &
```


4.2.11 Examples with Messages

4.2.11.1 Example 1

Merge 3 Presorted input files (fixed-length):

```
// JOB MERGE
// ASSGN SYS001,X'183' TAPE-OUTPUT
// ASSGN SYS002,X'131' DISK-INPUT-1
// ASSGN SYS013,X'180' TAPE-INPUT-2
// ASSGN SYS004,X'181' TAPE-INPUT-3
// DLBL SORTIN1,'AAAAAA',0
// EXTENT SYS002,,,0,60,60
// TLBL SORTIN3
// EXEC SORT
MERGE  FIELDS=(5,5,CH,A,15,4,FI,A),ORDER=3
RECORD LENGTH=100,TYPE=F
INPFIL BLKSIZE=2200
OUTFIL BLKSIZE=2200
OPTION LABEL=(U,S,U,S),SORTIN=(,13)
END
/*
/ &
```

Messages Printed

```
CASD069I NO SEVERE ERRORS
CASD201I RCD IN 0000450,OUT 0000450
CASD212I Unicenter CA-Sort END
```

Notes

- The output file and the input file TAPE-INPUT-2 are unlabeled tape files (variable U in LABEL operand).
- The LABEL operand could have been coded:

LABEL=(U,,U)

because S is the default value.

4.2.11.2 Example 2

Tag sort with fixed-length input and output on tape:

```
// JOB TAGSORT
// ASSGN SYS001,X'181' TAPE-OUTPUT
// ASSGN SYS002,X'131' DISK-INPUT
// ASSGN SYS003,X'133' SORT-WORK-AREA
// DLBL SORTIN1,'AAAAA',0
// EXTENT SYS002,,,0 100,120
// DLBL SORTWK1
// EXTENT SYS003,,,0,140,500
// EXEC SORT
  SORT    FIELDS=(29,3,PD,A,12,14,CH,D),WORK=1,SIZE=7000
  RECORD  TYPE=F,LENGTH=(107,27,27)
  INPFIL  BLKSIZE=3210
  OUTFIL  BLKSIZE=3483
  OPTION  ADDROUT=D,LABEL=U,STORAGE=30000
  END
/*
/ &
```

Messages Printed

```
CASD044I B= 03518
CASD045I G= 00344
CASD046I NMAX= 0106582
CASD071I STORAGE USED= 00030000
CASD069I NO SEVERE ERRORS
CASD201I RCD IN 0005000,OUT 0005000,ESTIM. 0007000
CASD210I END SORT PH, BIAS=00183
CASD222I 00023 TRACKS USED ON SORTWK1
CASD211I END MERGE PH
CASD201I RCD IN 0005000,OUT 0005000
CASD212I Unicenter CA-Sort END
```

4.2.11.3 Notes

- The LABEL operand could have been coded:

LABEL=(U,S)

As the rightmost value (S) is default, it does not have to be coded. With only one value (U) left, the parentheses may be omitted.

- The operand ADDROUT=D indicates that the output must include the input address as well as the control fields. The output record length is

input address + control field length¹ + control field length²

10 + 3 + 14 = 27

- The BIAS of this sort is 183 as shown in message CASD210I. The default value used by Unicenter CA-Sort is 150. An increase in performance may occur if the correct BIAS is coded in the SORT statement.

4.2.11.4 Example 3

Sort with variable-spanned records, VSE:

```
// JOB VSSORT
// ASSGN SYS001,X'131' DISK-OUTPUT
// ASSGN SYS002,X'131' DISK-INPUT
// ASSGN SYS003,X'133' SORT-WORK-AREA-1
// ASSGN SYS004,X'132' SORT-WORK-AREA-2
// DLBL SORTOUT,'AAAAA',0
// EXTENT SYS001,,,0,1200,360
// DLBL SORTIN1,'BBBBB',0
// EXTENT SYS002,,,0,500,400
// DLBL SORTWK1,,,0
// EXTENT SYS003,,,0,100,225
// DLBL SORTWK2,,,0
// EXTENT SYS004,,,0,200,225
// EXEC SORT,SIZE=64K
SORT   FIELDS=(5,10,CH,A,19,5,ZD,D),WORK=2,SIZE=8000
RECORD TYPE=VS,LENGTH=(604,,,40)
INPFIL BLKSIZE=3380
OUTFIL BLKSIZE=3400
END
/*
/&
```

Messages Printed

```
CASD032I L5= 00322
CASD044I B= 07136
CASD045I G= 000125
CASD046I NMAX= 0008651
CASD071I STORAGE USED = 00064000
CASD069I NO SEVERE ERRORS
CASD201I RCD IN 0004400,OUT 0004400,ESTIM. 0008000
CASD210I END SORT PH, BIAS=165
CASD222I 00205 TRACKS USED ON SORTWK1
CASD214I AVERAGE RL=00332
CASD211I END MERGE PH
CASD201I RCD IN 0004400,OUT 0004400
CASD212I Unicenter CA-Sort END
```

4.2.11.5 Notes

- Two sort work areas (SYS003 and SYS004) are used for this sort, with a total of 450 tracks. The estimated maximum number of records that can be sorted using these work areas is 8651 (message CASD046I).
- Because only 4400 records have to be sorted, only 205 of the 450 tracks of work area were used (message CASD222I).
- The modal length (record length that occurs most frequently) calculated by Unicenter CA-Sort is

$l^s = (\text{minimum record length} + \text{maximum record length}) / 2$ (rounded up to the next multiple of four).

The modal length in this example is 322 (message CASD032I).

$$l^s = \frac{40 + 604}{2} = 322$$

4.3 Year 2000 New Keyword Examples

The following examples illustrate the use of the new keywords (Y2C, Y2Z, Y2P, Y2D, PZ, PSI, ZSI) to handle the sorting and transforming of data representing date fields. These keywords read 80 byte image sysipt cards that have data representing date fields in different formats. They sort the data on the year according to the century window indicated by the Y2PAST= parameter on the OPTION statement. They also sort the data on the day and month in different formats. The format of the input cards is as follows:

Byte 1-6 Six-byte character date in YYMMDD format

Byte 7-8 Two-byte zoned decimal year data

Byte 9-10 Two-byte packed decimal year data

Byte 11-13

Three-byte packed decimal julian date in YYDDDS format

Byte 14-17

Four-byte packed decimal gregorian date in 0YYMMDDS format

Byte 18-80

Blanks

Each SORT statement sorts the year data in different formats. It also sorts the month and day data in different formats. Each OUTREC statement copies the first 20 bytes of the input record as is, inserts the word 'new data' into the output records, and then transforms the date data sorted on by the SORT statement according to the specified data format.

4.3.1.1 Example 1

The following example sorts 2-digit, 2-byte character year data in ascending order. OUTREC Y2C transforms the year data according to the century window.

```
// EXEC CA90SORT,SIZE=512K

COMPUTER ASSOCIATES          CA-SORT 9.0          06 APR 2002   15.00.48
CASD000I SORT  FIELDS=(1,2,Y2C,A),FILES=1
CASD000I OPTION PRINT=ALL,Y2PAST=79
CASD000I RECORD TYPE=F,LENGTH=80
CASD000I INPFIL SYSIPT
CASD000I OUTFIL SYSLST,DUMP
CASD000I OUTREC FIELDS=(1,20,C'NEW DATA',1,2,Y2C)
CASD000I END
CASD071I STORAGE USED= 00523520
CASD075I EFFECTIVE L3=00032
CADD001I ASSIGNED FILE=SORTWK1 SYS041=299 WORK01.
CASD105I CA-DYNAM/D PREOPEN PROCESSING FOR FILE SORTWK1
CASD116I CENTURY WINDOW= 1918-2017
CASD044I B= 46964
CASD045I G= 018974
CASD046I NMAX= 00008386
CASD069I NO SEVERE ERRORS
CASD201I RCD IN          0,OUT          16,ESTIM.          0
CASD202I RCD INSERT          16,DELETE          0
CASD210I END SORT PH, BIAS=NA
CASD218I IN-CORE SORT
```

RECORD LENGTH	CONTENT
32	1...5...10...15...20...25...30.. DEC 180430A8 NEW DATA1918 HEX FFFFFFFCF081100840444DCE4CCECFFFF 180430181F82C103C000556041311918
32	1...5...10...15...20...25...30.. DEC 180501A8 & NEW DATA1918 HEX FFFFFFFCF081110851444DCE4CCECFFFF 180501181F82F100C000556041311918
32	1...5...10...15...20...25...30.. DEC 61043001 / NEW DATA1961 HEX FFFFFFFDF016100140444DCE4CCECFFFF 610430616D12C603F000556041311961
32	1...5...10...15...20...25...30.. DEC 61050101 / & NEW DATA1961 HEX FFFFFFFDF016110151444DCE4CCECFFFF 610501616D12C600C000556041311961
32	1...5...10...15...20...25...30.. DEC 620430F2 NEW DATA1962 HEX FFFFFFFCF026100240444DCE4CCECFFFF 620430626F22F603C000556041311962
32	1...5...10...15...20...25...30.. DEC 620501F2 & NEW DATA1962 HEX FFFFFFFCF026110251444DCE4CCECFFFF 620501626F22C600F000556041311962
32	1...5...10...15...20...25...30.. DEC 990501R9 r & NEW DATA1999 HEX FFFFFFFDF099110951444DCE4CCECFFFF 990501999C92C900C000556041311999
32	1...5...10...15...20...25...30.. DEC 990430R9 r NEW DATA1999 HEX FFFFFFFDF099100940444DCE4CCECFFFF 990430999C92F903C000556041311999


```

32      1...5...10...15...20...25...30..
DEC 000501 0      &      NEW DATA2000
HEX FFFFFFFCF000110051444DCE4CCECFFFF
    000501000F02F000C000556041312000

32      1...5...10...15...20...25...30..
DEC 000430 0      NEW DATA2000
HEX FFFFFFFCF000100040444DCE4CCECFFFF
    000430000F02C003F000556041312000

```

RECORD LENGTH	CONTENT
32	1...5...10...15...20...25...30.. DEC 140430J4 < NEW DATA2014 HEX FFFFFFFDF041100440444DCE4CCECFFFF 140430141C42C103C000556041312014
32	1...5...10...15...20...25...30.. DEC 140501J4 < & NEW DATA2014 HEX FFFFFFFDF041110451444DCE4CCECFFFF 140501141C42F100F000556041312014
32	1...5...10...15...20...25...30.. DEC 150501A5) && NEW DATA2015 HEX FFFFFFFCF051110551444DCE4CCECFFFF 150501151D52C100C000556041312015
32	1...5...10...15...20...25...30.. DEC 150430A5) & NEW DATA2015 HEX FFFFFFFCF051100540444DCE4CCECFFFF 150430151D52C103C000556041312015
32	1...5...10...15...20...25...30.. DEC 170501J7 " & NEW DATA2017 HEX FFFFFFFDF071110751444DCE4CCECFFFF 170501171F72F100F000556041312017
32	1...5...10...15...20...25...30.. DEC 170430J7 " NEW DATA2017 HEX FFFFFFFDF071100740444DCE4CCECFFFF 170430171F72C103C000556041312017

4.3.1.2 Example 2

The following example sorts 2-digit, 2-byte zoned decimal year data in ascending order.
OUTREC Y2Z transforms year data according to the century window that is set.
OUTREC ZSI transforms any zoned decimal data.

```
// EXEC CA90SORT,SIZE=512K
```

```
COMPUTER ASSOCIATES                CA-SORT 9.0          06 APR 2002  15.00.52
CASD000I SORT  FIELDS=(7,2,Y2Z,A),FILES=1
CASD000I OPTION PRINT=ALL,Y2PAST=80
CASD000I RECORD TYPE=F,LENGTH=80
CASD000I INPFIL SYSIPT
CASD000I OUTFIL SYSLST,DUMP
CASD000I OUTREC FIELDS=(1,20,C'NEW DATA',7,2,Y2Z,7,2,ZSI)
CASD000I END
CASD071I STORAGE USED= 00523520
CASD075I EFFECTIVE L3=00034
CADD001I ASSIGNED FILE=SORTWK1 SYS041=299 WORK01.
CASD105I CA-DYNAM/D PREOPEN PROCESSING FOR FILE SORTWK1
CASD116I CENTURY WINDOW= 1917-2016
CASD044I B= 46964
CASD045I G= 018973
CASD046I NMAX= 00008386
CASD069I NO SEVERE ERRORS
CASD201I RCD IN          0,OUT          16,ESTIM.          0
CASD202I RCD INSERT          16,DELETE          0
CASD210I END SORT PH, BIAS=NA
CASD218I IN-CORE SORT
```

RECORD LENGTH	CONTENT
34	1...5...10...15...20...25...30...3 DEC 170501J7 " & NEW DATA191717 HEX FFFFFFFDF071110751444DCE4CCECFFFFFFF 170501171F72F100F00055604131191717
34	1...5...10...15...20...25...30...3 DEC 170430J7 " NEW DATA191717 HEX FFFFFFFDF0711100740444DCE4CCECFFFFFFF 170430171F72C103C00055604131191717
34	1...5...10...15...20...25...30...3 DEC 180430A8 NEW DATA191818 HEX FFFFFFFCF0811100840444DCE4CCECFFFFFFF 180430181F82C103C00055604131191818
34	1...5...10...15...20...25...30...3 DEC 180501A8 & NEW DATA191818 HEX FFFFFFFCF081110851444DCE4CCECFFFFFFF 180501181F82F100C00055604131191818
34	1...5...10...15...20...25...30...3 DEC 61043001 / NEW DATA196161 HEX FFFFFFFDF016100140444DCE4CCECFFFFFFF 610430616D12C603F00055604131196161
34	1...5...10...15...20...25...30...3 DEC 61050101 / & NEW DATA196161 HEX FFFFFFFDF016110151444DCE4CCECFFFFFFF 610501616D12C600C00055604131196161
34	1...5...10...15...20...25...30...3 DEC 620430F2 NEW DATA196262 HEX FFFFFFFCF026100240444DCE4CCECFFFFFFF 620430626F22F603C00055604131196262

```

34      1...5...10...15...20...25...30...3
DEC 620501F2      &      NEW DATA196262
HEX FFFFFFFCF026110251444DCE4CCECFFFFFFF
    620501626F22C600F00055604131196262

34      1...5...10...15...20...25...30...3
DEC 990501R9  r      &      NEW DATA199999
HEX FFFFFFFDF099110951444DCE4CCECFFFFFFF
    990501999C92C900C00055604131199999

34      1...5...10...15...20...25...30...3
DEC 990430R9  r      NEW DATA199999
HEX FFFFFFFDF099100940444DCE4CCECFFFFFFF
    990430999C92F903C00055604131199999

```

RECORD LENGTH	CONTENT
34	1...5...10...15...20...25...30...3 DEC 000501 0 & NEW DATA200000 HEX FFFFFFFCF000110051444DCE4CCECFFFFFFF 000501000F02F000C00055604131200000
34	1...5...10...15...20...25...30...3 DEC 000430 0 NEW DATA200000 HEX FFFFFFFCF000100040444DCE4CCECFFFFFFF 000430000F02C003F00055604131200000
34	1...5...10...15...20...25...30...3 DEC 140430J4 < NEW DATA201414 HEX FFFFFFFDF041100440444DCE4CCECFFFFFFF 140430141C42C103C00055604131201414
34	1...5...10...15...20...25...30...3 DEC 140501J4 < & NEW DATA201414 HEX FFFFFFFDF041110451444DCE4CCECFFFFFFF 140501141C42F100F00055604131201414
34	1...5...10...15...20...25...30...3 DEC 150501A5) && NEW DATA201515 HEX FFFFFFFCF051110551444DCE4CCECFFFFFFF 150501151D52C100C00055604131201515
34	1...5...10...15...20...25...30...3 DEC 150430A5) & NEW DATA201515 HEX FFFFFFFCF051100540444DCE4CCECFFFFFFF 150430151D52C103C00055604131201515

4.3.1.3 Example 3

The following example sorts 2-digit, 2-byte packed decimal year data in ascending order. OUTREC Y2P transforms year data according to the century window that is set.

```
// EXEC CA90SORT,SIZE=512K
```

```
COMPUTER ASSOCIATES                CA-SORT 9.0          06 APR 2002  15.00.54
CASD000I SORT  FIELDS=(9,2,Y2P,A),FILES=1
CASD000I OPTION PRINT=ALL,Y2PAST=81
CASD000I RECORD TYPE=F,LENGTH=80
CASD000I INPFIL SYSIPT
CASD000I OUTFIL SYSLST,DUMP
CASD000I OUTREC FIELDS=(1,20,C'NEW DATA',9,2,Y2P)
CASD000I END
CASD071I STORAGE USED= 00523520
CASD075I EFFECTIVE L3=00032
CADD001I ASSIGNED FILE=SORTWK1 SYS041=299 WORK01.
CASD105I CA-DYNAM/D PREOPEN PROCESSING FOR FILE SORTWK1
CASD116I CENTURY WINDOW= 1916-2015
CASD044I B= 46964
CASD045I G= 018974
CASD046I NMAX= 00008386
CASD069I NO SEVERE ERRORS
CASD201I RCD IN          0,OUT          16,ESTIM.          0
CASD202I RCD INSERT          16,DELETE          0
CASD210I END SORT PH, BIAS=NA
CASD218I IN-CORE SORT
```

RECORD LENGTH	CONTENT
32	1...5...10...15...20...25...30.. DEC 170501J7 " & NEW DATA1917 HEX FFFFFFFDF071110751444DCE4CCECFFFF 170501171F72F100F000556041311917
32	1...5...10...15...20...25...30.. DEC 170430J7 " NEW DATA1917 HEX FFFFFFFDF0711100740444DCE4CCECFFFF 170430171F72C103C000556041311917
32	1...5...10...15...20...25...30.. DEC 180430A8 NEW DATA1918 HEX FFFFFFFCF0811100840444DCE4CCECFFFF 180430181F82C103C000556041311918
32	1...5...10...15...20...25...30.. DEC 180501A8 & NEW DATA1918 HEX FFFFFFFCF081110851444DCE4CCECFFFF 180501181F82F100C000556041311918
32	1...5...10...15...20...25...30.. DEC 61043001 / NEW DATA1961 HEX FFFFFFFDF016100140444DCE4CCECFFFF 610430616D12C603F000556041311961
32	1...5...10...15...20...25...30.. DEC 61050101 / & NEW DATA1961 HEX FFFFFFFDF016110151444DCE4CCECFFFF 610501616D12C600C000556041311961
32	1...5...10...15...20...25...30.. DEC 620430F2 NEW DATA1962 HEX FFFFFFFCF026100240444DCE4CCECFFFF 620430626F22F603C000556041311962


```

32      1...5...10...15...20...25...30..
DEC 620501F2      &      NEW DATA1962
HEX FFFFFFFCF026110251444DCE4CCECFFFF
    620501626F22C600F000556041311962

32      1...5...10...15...20...25...30..
DEC 990501R9  r      &      NEW DATA1999
HEX FFFFFFFDF099110951444DCE4CCECFFFF
    990501999C92C900C000556041311999

32      1...5...10...15...20...25...30..
DEC 990430R9  r      NEW DATA1999
HEX FFFFFFFDF099100940444DCE4CCECFFFF
    990430999C92F903C000556041311999

```

RECORD LENGTH	CONTENT
32	1...5...10...15...20...25...30.. DEC 000501 0 & NEW DATA2000 HEX FFFFFFFCF000110051444DCE4CCECFFFF 000501000F02F000C000556041312000
32	1...5...10...15...20...25...30.. DEC 000430 0 NEW DATA2000 HEX FFFFFFFCF000100040444DCE4CCECFFFF 000430000F02C003F000556041312000
32	1...5...10...15...20...25...30.. DEC 140430J4 < NEW DATA2014 HEX FFFFFFFDF041100440444DCE4CCECFFFF 140430141C42C103C000556041312014
32	1...5...10...15...20...25...30.. DEC 140501J4 < & NEW DATA2014 HEX FFFFFFFDF041110451444DCE4CCECFFFF 140501141C42F100F000556041312014
32	1...5...10...15...20...25...30.. DEC 150501A5) && NEW DATA2015 HEX FFFFFFFCF051110551444DCE4CCECFFFF 150501151D52C100C000556041312015
32	1...5...10...15...20...25...30.. DEC 150430A5) & NEW DATA2015 HEX FFFFFFFCF051100540444DCE4CCECFFFF 150430151D52C103C000556041312015

4.3.1.4 Example 4

The following example illustrates the packed decimal julian date YYDDDS format. It sorts 2-digit, 1-byte decimal year data in ascending order. OUTREC Y2D transforms year data according to the century window that is set. OUTREC PSI transforms packed decimal data.

```
// EXEC CA90SORT,SIZE=512K

COMPUTER ASSOCIATES                CA-SORT 9.0                06 APR 2002  15.00.56
CASD000I SORT  FIELDS=(11,1,Y2D,A,12,2,PD,A),FILES=1
CASD000I OPTION PRINT=ALL,Y2PAST=90
CASD000I RECORD TYPE=F,LENGTH=80
CASD000I INPFIL SYSIPT
CASD000I OUTFIL SYSLST,DUMP
CASD000I OUTREC FIELDS=(1,20,C'NEW DATA',11,1,Y2D,12,2,PSI)
CASD000I END
CASD071I STORAGE USED= 00523520
CASD075I EFFECTIVE L3=00035
CADD001I ASSIGNED FILE=SORTWK1 SYS041=299 WORK01.
CASD105I CA-DYNAM/D PREOPEN PROCESSING FOR FILE SORTWK1
CASD116I CENTURY WINDOW= 1915-2014
CASD044I B= 46964
CASD045I G= 018973
CASD046I NMAX= 00008386
CASD069I NO SEVERE ERRORS
CASD201I RCD IN          0,OUT          16,ESTIM.          0
CASD202I RCD INSERT          16,DELETE          0
CASD210I END SORT PH, BIAS=NA
CASD218I IN-CORE SORT
```

RECORD LENGTH	CONTENT
35	1...5...10...15...20...25...30...35 DEC 150430A5) & NEW DATA1915120 HEX FFFFFFFCF051100540444DCE4CCECFFFFFFF 150430151D52C103C000556041311915120
35	1...5...10...15...20...25...30...35 DEC 150501A5) && NEW DATA1915121 HEX FFFFFFFCF051110551444DCE4CCECFFFFFFF 150501151D52C100C000556041311915121
35	1...5...10...15...20...25...30...35 DEC 170430J7 " NEW DATA1917120 HEX FFFFFFFDF071100740444DCE4CCECFFFFFFF 170430171F72C103C000556041311917120
35	1...5...10...15...20...25...30...35 DEC 170501J7 " & NEW DATA1917121 HEX FFFFFFFDF071110751444DCE4CCECFFFFFFF 170501171F72F100F000556041311917121
35	1...5...10...15...20...25...30...35 DEC 180430A8 NEW DATA1918120 HEX FFFFFFFCF081100840444DCE4CCECFFFFFFF 180430181F82C103C000556041311918120
35	1...5...10...15...20...25...30...35 DEC 180501A8 & NEW DATA1918121 HEX FFFFFFFCF081110851444DCE4CCECFFFFFFF 180501181F82F100C000556041311918121
35	1...5...10...15...20...25...30...35 DEC 61043001 / NEW DATA1961120 HEX FFFFFFFDF016100140444DCE4CCECFFFFFFF 610430616D12C603F000556041311961120
35	1...5...10...15...20...25...30...35 DEC 61050101 / & NEW DATA1961121 HEX FFFFFFFDF016110151444DCE4CCECFFFFFFF 610501616D12C600C000556041311961121

```

35      1...5...10...15...20...25...30...35
DEC 620430F2          NEW DATA1962120
HEX FFFFFFFCF02610024044DCE4CCECFFFFFFF
    620430626F22F603C000556041311962120

35      1...5...10...15...20...25...30...35
DEC 620501F2          &    NEW DATA1962121
HEX FFFFFFFCF026110251444DCE4CCECFFFFFFF
    620501626F22C600F000556041311962121

```

RECORD LENGTH	CONTENT
35	1...5...10...15...20...25...30...35 DEC 990430R9 r NEW DATA1999120 HEX FFFFFFFDF099100940444DCE4CCECFFFFFFF 990430999C92F903C000556041311999120
35	1...5...10...15...20...25...30...35 DEC 990501R9 r & NEW DATA1999121 HEX FFFFFFFDF099110951444DCE4CCECFFFFFFF 990501999C92C900C000556041311999121
35	1...5...10...15...20...25...30...35 DEC 000430 0 NEW DATA2000120 HEX FFFFFFFCF000100040444DCE4CCECFFFFFFF 000430000F02C003F000556041312000120
35	1...5...10...15...20...25...30...35 DEC 000501 0 & NEW DATA2000121 HEX FFFFFFFCF000110051444DCE4CCECFFFFFFF 000501000F02F000C000556041312000121
35	1...5...10...15...20...25...30...35 DEC 140430J4 < NEW DATA2014120 HEX FFFFFFFDF041100440444DCE4CCECFFFFFFF 140430141C42C103C000556041312014120
35	1...5...10...15...20...25...30...35 DEC 140501J4 < & NEW DATA2014121 HEX FFFFFFFDF041110451444DCE4CCECFFFFFFF 140501141C42F100F000556041312014121

4.3.1.5 Example 5

The following example illustrates packed decimal date in 0YYMMDDS format. It sorts 2-digit, 2-byte packed decimal year data in ascending order. OUTREC Y2P transforms the year data according to the century window that is set. OUTREC PZ transforms partially packed decimal data.

```
// EXEC CA90SORT,SIZE=512K
```

```
COMPUTER ASSOCIATES                CA-SORT 9.0          06 APR 2002  15.00.58
CASD000I SORT  FIELDS=(14,2,Y2P,A,15,2,PD0,A,16,2,PD0,A),FILES=1
CASD000I OPTION PRINT=ALL,Y2PAST=83
CASD000I RECORD TYPE=F,LENGTH=80
CASD000I INPFIL SYSIPT
CASD000I OUTFIL SYSLST,DUMP
CASD000I OUTREC FIELDS=(1,20,C'NEW DATA',14,2,Y2P,15,2,PZ,16,2,PZ)
CASD000I END
CASD071I STORAGE USED= 00523520
CASD075I EFFECTIVE L3=00036
CADD001I ASSIGNED FILE=SORTWK1 SYS041=299 WORK01.
CASD105I CA-DYNAM/D PREOPEN PROCESSING FOR FILE SORTWK1
CASD116I CENTURY WINDOW= 1914-2013
CASD044I B= 46964
CASD045I G= 018255
CASD046I NMAX= 00008386
CASD069I NO SEVERE ERRORS
CASD201I RCD IN          0,OUT          16,ESTIM.          0
CASD202I RCD INSERT          16,DELETE          0
CASD210I END SORT PH, BIAS=NA
CASD218I IN-CORE SORT
```

RECORD LENGTH	CONTENT
36	1...5...10...15...20...25...30...35. DEC 140430J4 < NEW DATA19140430 HEX FFFFFFFDF041100440444DCE4CCECFFFFFFFF 140430141C42C103C0005560413119140430
36	1...5...10...15...20...25...30...35. DEC 140501J4 < & NEW DATA19140501 HEX FFFFFFFDF041110451444DCE4CCECFFFFFFFF 140501141C42F100F0005560413119140501
36	1...5...10...15...20...25...30...35. DEC 150430A5) & NEW DATA19150430 HEX FFFFFFFCF051100540444DCE4CCECFFFFFFFF 150430151D52C103C0005560413119150430
36	1...5...10...15...20...25...30...35. DEC 150501A5) && NEW DATA19150501 HEX FFFFFFFCF051110551444DCE4CCECFFFFFFFF 150501151D52C100C0005560413119150501
36	1...5...10...15...20...25...30...35. DEC 170430J7 " NEW DATA19170430 HEX FFFFFFFDF071100740444DCE4CCECFFFFFFFF 170430171F72C103C0005560413119170430
36	1...5...10...15...20...25...30...35. DEC 170501J7 " & NEW DATA19170501 HEX FFFFFFFDF071110751444DCE4CCECFFFFFFFF 170501171F72F100F0005560413119170501
36	1...5...10...15...20...25...30...35. DEC 180430A8 NEW DATA19180430 HEX FFFFFFFCF081100840444DCE4CCECFFFFFFFF 180430181F82C103C0005560413119180430
36	1...5...10...15...20...25...30...35. DEC 180501A8 & NEW DATA19180501 HEX FFFFFFFCF081110851444DCE4CCECFFFFFFFF 180501181F82F100C0005560413119180501


```

36      1...5...10...15...20...25...30...35.
DEC 61043001 /          NEW DATA19610430
HEX FFFFFFFDF016100140444DCE4CCECFFFFFFFF
    610430616D12C603F0005560413119610430

36      1...5...10...15...20...25...30...35.
DEC 61050101 /      &    NEW DATA19610501
HEX FFFFFFFDF016110151444DCE4CCECFFFFFFFF
    610501616D12C600C0005560413119610501

```

RECORD LENGTH	CONTENT
36	1...5...10...15...20...25...30...35. DEC 620430F2 NEW DATA19620430 HEX FFFFFFFCF026100240444DCE4CCECFFFFFFFF 620430626F22F603C0005560413119620430
36	1...5...10...15...20...25...30...35. DEC 620501F2 & NEW DATA19620501 HEX FFFFFFFCF026110251444DCE4CCECFFFFFFFF 620501626F22C600F0005560413119620501
36	1...5...10...15...20...25...30...35. DEC 990430R9 r NEW DATA19990430 HEX FFFFFFFDF099100940444DCE4CCECFFFFFFFF 990430999C92F903C0005560413119990430
36	1...5...10...15...20...25...30...35. DEC 990501R9 r & NEW DATA19990501 HEX FFFFFFFDF099110951444DCE4CCECFFFFFFFF 990501999C92C900C0005560413119990501
36	1...5...10...15...20...25...30...35. DEC 000430 0 NEW DATA20000430 HEX FFFFFFFCF000100040444DCE4CCECFFFFFFFF 000430000F02C003F0005560413120000430
36	1...5...10...15...20...25...30...35. DEC 000501 0 & NEW DATA20000501 HEX FFFFFFFCF000110051444DCE4CCECFFFFFFFF 000501000F02F000C0005560413120000501

4.4 Initiation from an Executing Program

The user's program (written in assembler language) may initiate Unicenter CA-Sort execution by a LOAD macro instruction for the first Unicenter CA-Sort module (same name as appears on the // EXEC statement). The loading address may be defined anywhere in the partition. The package then occupies the storage space between this address and the highest partition limit (unless the STORAGE option is present). Standard VSE linkage conventions must be used and the necessary sort data and parameters must be passed as follows to the Unicenter CA-Sort system:

Register	Contents
1	This register must contain the address of a list of required sort parameters.
13	This must contain the address of a nine-doubleword area in which the sort/merge program can save the contents of the user's registers. The user's registers will be restored when Unicenter CA-Sort completes its processing.
14	This register must contain the address in the user's coding to which Unicenter CA-Sort will return control upon completion of sort/merge functions.
15	This register must contain the entry point (load) address.

4.4.1 Parameter List

Register 1 must contain the address of a list of at least 10 address constants. The format of this list is shown on the next page. The first six and the last four constants are the addresses of sort/merge control statement images, which take the place of control statements. Any parameter, except for the first two and the tenth, may be assigned a null value by means of a zero address constant (for example, A(0)). The first parameter may be defined as a zero address constant only if one or more address pointers to SORTx statements are provided elsewhere in the list.

The parameter list will be generated by the COBOL compiler if the SORT verb is used. Otherwise, the parameters must be created by the user in the following format: operation definier (for example, SORT), one blank, operands, one blank. No extra blanks, continuation characters, statement sequence numbers, comments, and so forth, are allowed in these parameters. Below is an example of the coding necessary to create a SORT control statement image. The address of the beginning of this statement would be placed in the first entry in the parameter list.

```
DC      C'SORT FIELDS=(1,24,CH,A,46,4,CH,D), '  
DC      C'WORK=2 '
```

Three address constants contain addresses of branch tables for the preloaded user routines of each phase. These routines may be part of the program that initiates the Unicenter CA-Sort program. See the chapter entitled "Loading User Exit Routines" for information on preloaded user routines.

The next address constant in the parameter list is the address of a halfword that will be used by Unicenter CA-Sort to return a code to the initiating program. This code will be 0 if the sort/merge program successfully completes execution, or 16 if execution is unsuccessful. This is the only parameter in the user's program that is modified by the sort/merge program (required address constant).

Register 1

ADDR of PARMLIST



PARMLIST

DC	A(SORT or MERGE statement)
DC	A(RECORD statement)
DC	A(INPFIL statement)
DC	A(OUTFIL statement)
DC	A(OPTION statement)
DC	A(MODS statement)
DC	A(Branch table for PH-1 preloaded user routines)
DC	A(Branch table for PH-2 preloaded user routines)
DC	A(Branch table for PH-3 preloaded user routines)
DC	A(Return code halfword)
DC	A(ALTSEQ statement or AQT constant)
DC	A(OUTREC statement)
DC	A(SUM statement)
DC	A(INCLUDE/OMIT statement)
DC	A(ANALYZE statement)
DC	A(SORT1 statement)
DC	A(SORT2 statement)
:	:
DC	A(SORT9 statement)
DC	A(INREC statement)

The ALTSEQ address constant contains zero if no changes in the standard EBCDIC collating sequence are necessary. If alternate sequence is requested, this address constant points to the beginning of the ALTSEQ statement image (same coding rules as for SORT, RECORD, and so forth). Optionally, the user may specify a 256-byte translation table directly, using the following linkages:

	DC	A(SORT or MERGE statement)	
	DC	A(RECORD statement)	
	.		
	DC	A(Return code halfword)	
	DC	A(AQTT)	ALTSEQ entry pointer
	/	.	
AQTT	DC	C'AQTT'	AQTT constant and translation
	DC	A(AQTAB)	table pointer
	/	.	
AQTAB	DC	X'0001020304050607'	256-byte
	DC	X'08090A0B0C0D0E0F'	user-built EBCDIC translation
			table for control fields
	*		
	*		
	DC	X'F8F9FAFBFCFDFEFF'	

If Unicenter CA-Sort has been invoked from another program, it is the user's responsibility to check the return code upon receiving control back after the sort to ensure that the execution completed normally. In COBOL, this is easily accomplished by testing the reserved register, SORT-RETURN. A nonzero value indicates that a fatal error has occurred. Additional messages will be found on the console and the printer.

4.4.2 Examples

Example of an assembler language program invoking Unicenter CA-Sort with control passed to preloaded exit routines:

```

INVOKED  START 0
          BALR 12,0          .LOAD BASE REGISTER
          USING *,12         .ESTABLISH ADDRESSABILITY
          LOAD SORT,ENDINV   .LOAD Unicenter CA-Sort AFTER THE MAIN PROGRAM
          LR 15,1            .REG 15 = ENTRY POINT OF Unicenter CA-Sort
          LA 1,PARM          .REG 1 = ADDR OF SORT PARAMETERS
          LA 13,SAVAR        .REG 13 = ADDR OF REGISTER SAVE AREA
          BALR 14,15         .PASS CONTROL TO Unicenter CA-Sort

*
* Unicenter CA-Sort WILL RETURN CONTROL AT THIS POINT
*
          LH 15,RTRNCD       .LOAD THE RETURN CODE FROM Unicenter CA-Sort
          LTR 15,15          .CHECK FOR NORMAL COMPLETION
          BNZ SORTERR        .ERROR IF NON-ZERO
FINISH    EQU *              .ELSE NORMAL EOJ OR OTHER PROCESSING
          EOJ
SORTERR   EQU *              .ERROR DURING THE SORT/MERGE
          DUMP               .TERMINATE ABNORMALLY
SAVAR     DS 9D              .REGISTER SAVE AREA
=====
* SORT/MERGE CONTROL CARD PARAMETER LIST PASSED TO Unicenter CA-Sort *
=====
PARM      DC A(SRTSTMT)      .POINTER TO SORT STATEMENT
          DC A(RCDSTMT)      .POINTER TO RECORD STATEMENT
          DC A(INPSTMT)      .POINTER TO INPFIL STATEMENT
          DC A(OUTSTMT)      .POINTER TO OUTFIL STATEMENT
          DC A(OPTSTMT)      .POINTER TO OPTION STATEMENT
          DC A(MODSTMT)      .POINTER TO MODS STATEMENT
          DC A(PH1EXIT)      .POINTER TO PHASE-1 EXIT ROUTINES
          DC A(0)            .NO PHASE-2 EXITS
          DC A(PH3EXIT)      .POINTER TO PHASE-3 EXIT ROUTINES
          DC A(RTRNCD)       .POINTER TO RETURN CODE
          DC A(0)            .ALTSEQ NOT USED
          DC A(0)            .OUTREC NOT USED
          DC A(0)            .SUM NOT USED
          DC A(0)            .INCL/OMIT NOT USED
          DC A(0)            .ANALYZE NOT USED

*
SRTSTMT   DC C'SORT FIELDS=(5,15,BI,A),WORK=1 ' .SORT STMT IMAGE
RCDSTMT   DC C'RECORD TYPE=F,LENGTH=100 ' .RECORD STMT IMAGE
INPSTMT   DC C'INPFIL BLKSIZE=2200,VOLUME=2 ' .INPFIL STMT IMAGE
OUTSTMT   DC C'OUTFIL BLKSIZE=2200 ' .OUTFIL STMT IMAGE
OPTSTMT   DC C'OPTION LABEL=(N,N),STORAGE=98K ' .OPTION STMT IMAGE
MODSTMT   DC C'MODS PH1=(,E11,E17),PH3=(,E31,E37) ' .MODS STMT
*
RTRNCD    DC H'0'           .RETURN COMPLETION CODE
          LTORG              .STRAY LITERALS HERE

```

```

=====
*          PRELOADED EXIT ROUTINES FOR PHASE 1 PROCESSING          *
=====
PH1EXIT EQU *
        USING *,15                      .REG 15 = BASE REGISTER UPON ENTRY
        B      E11RTN                  .POINTER TO E11 ROUTINE FOR HDR LBLS
        DC     A(0)                    .E15 NOT USED
        B      E17RTN                  .POINTER TO E17 ROUTINE FOR TLR LBLS
        DC     A(0)                    .E18 NOT USED

*
E11RTN  SAVE (14,12)                  .SAVE REGISTERS UPON ENTRY
        :      :                      :
        :      :                      :
        RETURN (14,12)                .RETURN TO Unicenter CA-Sort

*
E17RTN  SAVE (14,12)                  .SAVE REGISTERS UPON ENTRY
        :      :                      :
        :      :                      :
        RETURN (14,12)                .RETURN TO Unicenter CA-Sort

=====
*          PRELOADED EXIT ROUTINES FOR PHASE 3 PROCESSING          *
=====
PH3EXIT EQU *
        USING *,15                      .REG 15 = BASE REGISTER UPON ENTRY
        B      E31RTN                  .POINTER TO E31 ROUTINE FOR HDR LBLS
        DC     A(0)                    .E32 NOT USED
        DC     A(0)                    .E35 NOT USED
        B      E37RTN                  .POINTER TO E37 ROUTINE FOR TLR LBLS
        DC     A(0)                    .E38 NOT USED
        DC     A(0)                    .E39 NOT USED

*
E31RTN  SAVE (14,12)                  .SAVE REGISTERS UPON ENTRY
        :      :                      :
        :      :                      :
        RETURN (14,12)                .RETURN TO Unicenter CA-Sort

*
E37RTN  SAVE (14,12)                  .SAVE REGISTERS UPON ENTRY
        :      :                      :
        :      :                      :
        RETURN (14,12)                .RETURN TO Unicenter CA-Sort

=====
*          END OF MAIN PROGRAM -- LOAD POINT FOR Unicenter CA-Sort          *
=====
ENDINV  DS     0D                      .ALIGN ON A DOUBLEWORD BOUNDARY
        END      INVOKED

```


Example of a COBOL Program Initiating the Sort/Merge

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  EXAMPLE.
AUTHOR.     COMPUTER ASSOCIATES.
REMARKS.    SAMPLE COBOL PROGRAM WHICH USES THE 'SORT' VERB
            TO READ A DISK FILE IN, SORT IT AND WRITE OUT
            ANOTHER DISK FILE.  THIS PARTICULAR EXAMPLE
            ILLUSTRATES THE FOLLOWING OPTIONS RELATED TO THE
            COBOL 'SORT' VERB:
                1) USING CLAUSE
                2) OUTPUT PROCEDURE CLAUSE
                3) HOW THE WORK FILE IS DEFINED IN THE DATA
                   DIVISION (AN 'SD' ENTRY)
                4) HOW AN 'OPTION' CONTROL STATEMENT CAN BE
                   PASSED FROM A COBOL PROGRAM TO SORT
                5) HOW THE RETURN CODE FROM SORT IS CHECKED
                   TO INSURE SUCCESSFUL COMPLETION

ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
    SELECT IN-FILE    ASSIGN TO SYS012-DA-3350-S-INFILE.
    SELECT OUT-FILE   ASSIGN TO SYS013-DA-3350-S-OUTFILE.
    SELECT SORT-FILE  ASSIGN TO SYS001-DA-3350-S-SORTWK1.

DATA DIVISION.
FILE SECTION.
FD  IN-FILE
    LABEL RECORDS ARE STANDARD
    RECORDING MODE IS F
    DATA RECORDS ARE IN-REC.
01  IN-REC      PIC X(80).
*
FD  OUT-FILE
    LABEL RECORDS ARE STANDARD
    RECORDING MODE IS F
    DATA RECORDS ARE OUT-REC.
01  OUT-REC     PIC X(80).
*
SD  SORT-FILE
    LABEL RECORDS ARE STANDARD
    RECORDING MODE IS F
    SORT-OPTION IS OPTION-STMT
    DATA RECORDS ARE SORT-REC.
01  SORT-REC.
    05  SR-KEY-1    PIC X(10).
    05  FILLER      PIC X(20).
    05  SR-KEY-2    PIC X(8).
    05  FILLER      PIC X(42).
```

```

WORKING-STORAGE SECTION.
01 OPTION-STMT    PIC X(12)          VALUE 'OPTION EQUALS '.
01 EOF-SW         PIC S9             COMP-3  VALUE ZERO.
   88 AT-EOF      VALUE +1.
01 COUNTER        PIC S9(5)          COMP-3  VALUE ZERO.
01 WS-REC.
   05 WS-15       PIC X(15).
   05 FILLER      PIC X(65).

PROCEDURE DIVISION.
MAINLINE.
   SORT SORT-FILE ON ASCENDING KEY SR-KEY-1, SR-KEY-2
   USING IN-FILE
   OUTPUT PROCEDURE IS SRT-OUT.

   IF SORT-RETURN NOT EQUAL ZERO
      DISPLAY 'ATTENTION - SORT FAILED' UPON CONSOLE
      CALL 'ABEND'
   ELSE
      DISPLAY 'TOTAL RECORDS SORTED= ' COUNTER
      CLOSE OUT-FILE
      STOP RUN.

*
SRT-OUT SECTION.
SRT-OUT-MAINLINE.
   OPEN OUTPUT OUT-FILE
   PERFORM RETURN-SORT-RECS
   UNTIL AT-EOF.

*
SRT-OUT-ROUTINES SECTION.
RETURN-SORT-RECS.
   RETURN SORT-FILE INTO WS-REC
   AT END MOVE 1 TO EOF-SW.
   IF AT-EOF
      NEXT SENTENCE
   ELSE
      ADD 1 TO COUNTER
   WRITE OUT-REC FROM WS-REC
   INVALID KEY CALL 'ABEND'.

```

4.5 Unicenter CA-Sort Extended Features

4.5.1 Unicenter CA-Sort and Unicenter CA-Dynam/D Interface

The Unicenter CA-Sort and Unicenter CA-Dynam/D interface is provided for Unicenter CA-Sort users who also have the Computer Associates software product, Unicenter CA-Dynam/D. Unicenter CA-Dynam/D is a disk and catalog management system implemented for the VSE user. The interface is activated by specifying DYNAMD=Y in the CASORT generation option macro or on the OPTION control statement. The interface accomplishes the following:

- Sort work files are always deleted from the VTOC whenever the sorted output file is closed. When the IBM sort/merge is executing, the sort work files are deleted from the VTOC only when the file SORTOUT is closed. For example, this would occur when another program invokes the IBM sort from a high-level language and does not define a file by the name of SORTOUT. In that case, the sort work file will not be deleted.
- The sorted output file will always be truncated unless otherwise specified. All unused space allocated for the file will be released when the file is closed.
- The sort work files will be truncated if possible before the SORTOUT file is opened, releasing all unused allocated space.

When running Unicenter CA-Sort with Unicenter CA-Dynam/D Release 5.3 or above, the new interface accomplishes the following in addition to those already listed:

- Provides Unicenter CA-Sort with the input file device type during Phase 0 processing.
- Provides Unicenter CA-Sort with the VTOC CISIZE if input is a SAM FBA file during Phase 0 processing.
- Provides Unicenter CA-Sort with the input file attributes, if available in the Unicenter CA-Dynam catalog or Unicenter CA-Dynam/D-created FORMAT 1 DSCB during Phase 0 processing. The attributes are
 - logical record length
 - record format
 - block size
 - file size estimate.

The file information is not supplied by Unicenter CA-Dynam/D if already done so by the Unicenter CA-Dynam/FI interface. The Unicenter CA-Dynam/FI interface, if activated, is called prior to calling Unicenter CA-Dynam/D (Message CASD104I will be issued when Unicenter CA-Dynam/FI is called. Message CASD105I will be issued when the new Unicenter CA-Dynam/D interface is called.) Any or all of the file parameters supplied by Unicenter CA-Dynam/D may be overridden by specifying the corresponding parameters in the Unicenter CA-Sort control statements.

For files created under Unicenter CA-Dynam/D control using DTFPH, the file information may not be correct and must be provided by the user.

- In addition, care must be taken in the use of the option FILNM parameter when using Unicenter CA-Dynam/D generation (GDG) data sets. It is restricted to use option FILNM to access a generation data set for input and output using the same DTFNAME for both files.
- Provides Unicenter CA-Sort with the output file device type during Phase 0 processing. Unicenter CA-Sort will then call the Unicenter CA-Dynam/FI interface, if active, to perform block size, CISIZE optimization (if requested by user) based on the output file disk device type supplied by Unicenter CA-Dynam/D and the logical record length determined by Unicenter CA-Sort.
- Updates the Unicenter CA-Dynam catalog and FORMAT 1 DSCB, if applicable to the output file attributes: logical record length, record format, block size, CISIZE, and so forth.

Note: For proper block size optimization, the record format and length must be made available to the Unicenter CA-Dynam/FI interface through either the sort record statement, the Unicenter CA-Dynam catalog, or Unicenter CA-Dynam/FI JCL parameters and cannot be obtained by the Unicenter CA-Dynam/D open interface, since it is called after the Unicenter CA-Dynam/FI interface.

- Performs dynamic allocation of work files. If no work file labels are found during Unicenter CA-Sort Phase 0 processing, and the CASORT option macro parameter DYNALOC=(volser,n) has been specified at generation time, dynamic allocation will be initiated (unless WORK=0 is specified on the SORT control statement). The Unicenter CA-Dynam/D interface is called to add the necessary SORTWK_n labels and to generate a temporary FILEID for each SORTWK. Subsequent OPEN, truncate, CLOSE, and delete processings are the same as for user-defined work files by JCL statements.

Where will the work files be allocated?

On the volumes specified in the DYNALOC=volser sublist parameter. The volser may be up to six characters and may be any valid DASD volume serial number, a DYNAM/D generic volser, or disk pool-ID.

How many work files will be allocated?

As many as specified in the DYNALOC=n sublist parameter. n may be in the range 1-9. If n is omitted, three work files will be allocated.

How much DASD space will be allocated?

The total amount of DASD space to allocate for all the work files is determined as follows:

- If no file size estimates are given by the user by the SIZE=n parameter on the SORT control statement or by the Unicenter CA-Dynam interface, the default values will be used:

block size = 1000

number of blocks = 6000

Thus, 6000 blocks, each containing 1000 bytes, will be allocated.

- If file size estimates are given when using fixed-length records, the maximum allocation is as follows:

$$\text{RPB} = (c + l^2 - 1) \text{MOD } l^2$$

$$\text{block size} = \text{RPB} * l^2$$

$$\text{number of blocks} = \text{file size (in no. of records)} / \text{RPB} + 1$$

where l^2 is the internal sort record length after INREC and E15 record modification.

c is one of the following (based on the lowest capacity work file device specified on the DISK= parameter at generation time):

DISK=	c
2311	822
2314	1084
3330	2050
3340	1247
3350	3016
3375	5592
3380	6356
3390	10796
3310	2552
3370	2552
93xx	2552

- If file size estimates are given when using variable-length records, the maximum allocation is as follows:

$$\text{file size (in bytes)} = \text{file size (in no. of records)} * l^5$$

$$\text{block size} = c$$

$$\text{number of blocks} = \text{file size (in bytes)} / \text{block size} + 1$$

where l^5 is the modal record length for variable-length records. It is recommended that an accurate l^5 value be specified by the user for work file space allocation. (See 'RECORD Control Statement' for more information on l^5 length.)

The total amount of DASD space to allocate will be distributed evenly across the number of work files defined.

4.5.2 Unicenter CA-Sort and Unicenter CA-Dynam/FI Interface

The Unicenter CA-Sort program is capable of interfacing to Unicenter CA-Dynam/FI in order to take advantage of Unicenter CA-Dynam/FI specified file characteristics. The interface is activated by specifying DYNAMFI=Y in either the CASORT option generation macro or on the OPTION control statement. The Unicenter CA-Dynam/FI user may specify values on either job control DLBL or TLBL statements or in the Unicenter CA-Dynam/FI catalog which represent any of the following file attributes:

- sort/merge control fields (keys)
- block size
- logical record length
- record format
- logical unit
- standard or unlabeled tapes
- tape positioning
- number of concatenated files

The values listed above will be used by Unicenter CA-Sort to override values specified in the sort/merge control statements. At the end of the sort/merge, Unicenter CA-Dynam/FI will be called again to cause the catalog to be updated with file information obtained during execution.

Unicenter CA-Dynam/FI Release 5.0 (or higher) users may optionally place, in the catalog, sort/merge control fields to be retrieved during sort execution using a symbolic name. For example:

```
SORT KEY=PAYROLL
```

will retrieve a set of sort/merge keys stored in the catalog using Unicenter CA-Dynam/D catalog utilities, for use with the user's payroll processing. The KEY parameter may be used on the SORT, MERGE and SORTx control statements if Release 5.0 or higher of Unicenter CA-Dynam/FI is installed and active.

If sufficient information is stored in the catalog concerning the record format, logical record lengths and block sizes of the files being sorted, Unicenter CA-Dynam/FI users may sort or merge a file with only the SORT/MERGE/SORTx control statement. This allows Unicenter CA-Dynam/FI to supply the information which would normally be placed on the RECORD, INPFIL and OUTFIL control statements.

The following Unicenter CA-Dynam/FI keywords are applicable to Unicenter CA-Sort usage:

BLKSZ
LRECL
RECFM
SYSxxx
VERIFY
RUN
SKR
NORWD
SL
UL

Limited support for concatenation of input files read by Unicenter CA-Sort is provided by the Unicenter CA-Sort and Unicenter CA-Dynam/FI interface. Unicenter CA-Dynam/FI passes the number of input files to be concatenated to Unicenter CA-Sort and subsequently performs label and OPEN/CLOSE processing for the concatenated files. All files to disk are stored in SORTIN1 concatenation. All files to tape are stored in SORTIN2 concatenation.

4.5.2.1 Notes on the Catalog Information

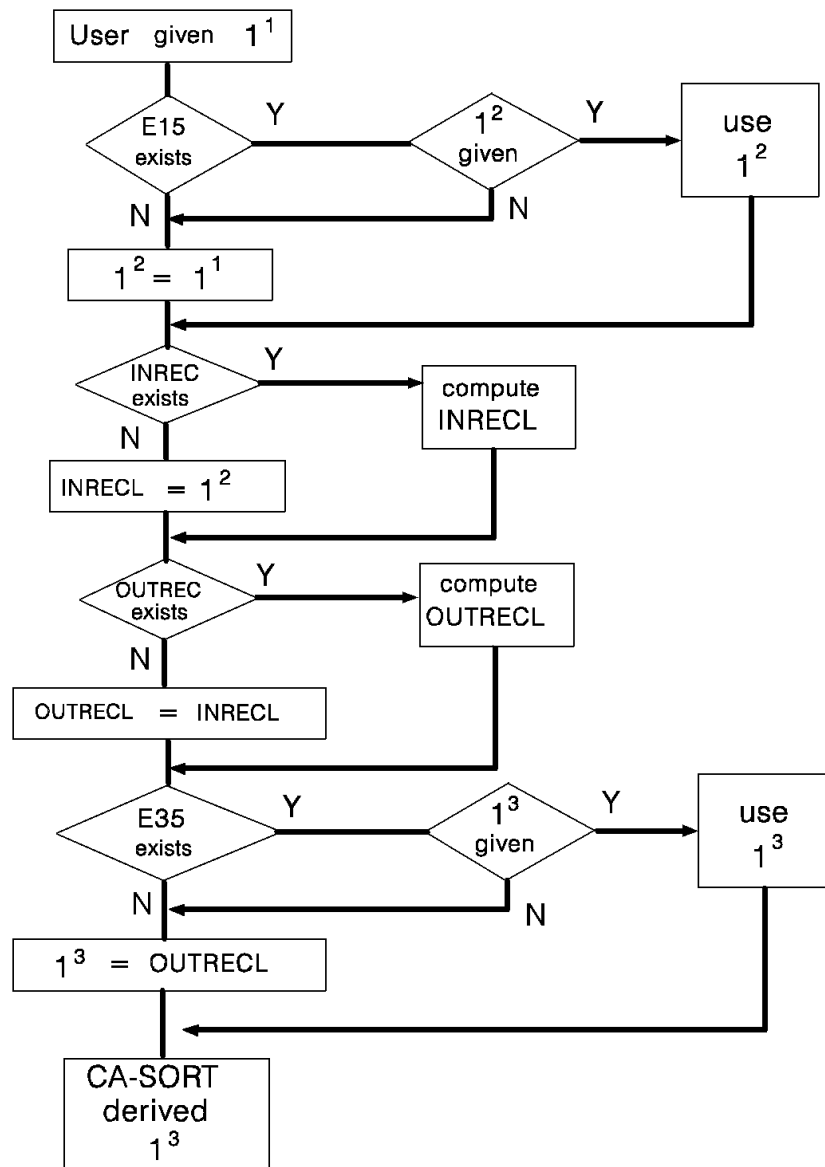
- The record formats (RECFM) must be the same for SORTINn and SORTOUT files.
- The logical record lengths (LRECL), if defined to Unicenter CA-Dynam/FI, must be the same for SORTINn and SORTOUT files, unless record reformatting is to be performed by Unicenter CA-Sort.
- If record reformatting is to be performed by Unicenter CA-Sort, it is recommended that the output file logical record length not be predefined to Unicenter CA-Dynam/FI. Unicenter CA-Sort will call the Unicenter CA-Dynam/FI interface update routine with the derived output record length l³ when creating the output file. See the flowchart on the next page.
- The following will result in Unicenter CA-Sort message CASD078W:
 - l³ values disagree and no E35 exit exists.
 - E35 exit exists, but the l³ value returned by the Unicenter CA-Dynam/FI interface disagrees with the value specified by the user.

Example

```
// TLBL SORTOUT, 'SORT-OUTPUT', BLKSZ=800
// EXEC SORT
//   SORT   FIELDS= ...
//   RECORD TYPE=F, LENGTH=80
//   OUTFIL BLKSIZE=80
//   END
/*
```

In the above example, the block size for the output file is changed from 80 to 800 by specifying the Unicenter CA-Dynam/FI BLKSZ keyword.

Derive 1^3 based on 1^1 and other user-specified record processing options as follows:



4.5.3 Unit Record Input and Output

Unit record input and output processings may be initiated by the specification of one of the following keywords on the INPFIL or OUTFIL control statements:

- SYSIPT
- SYSLST
- PUNCH
- PCHLST

See the sections in the chapter entitled "Using the Control Language" on these two control statements for complete details.

If the BLKSIZE= keyword is placed on the INPFIL or OUTFIL control statement along with the SYSIPT/SYSLST/PUNCH/PCHLST keywords, unit record processing will be done in addition to normal SORTINn and SORTOUT file processing. If the BLKSIZE= parameter has not been specified, Unicenter CA-Sort will assume that only unit record operations are to be performed for the respective input or output phase of execution.

Existing MODS control statements which still reference the unit record E15 (CAIE15 statement read) and E35 (CAPRINT) exits will continue to function. However, it is strongly recommended that they be changed and to use the INPFIL and OUTFIL unit record options listed above. CAIE15 is no longer distributed with Unicenter CA-Sort.

Example of Reading Input Data from SYSIPT

The following job control statements show how the control statements should be defined to read the SYSIPT input file. Note that the default logical unit used for the first work file is SYS002 when SORTIN is SYSIPT and no other SORTIN files are to be processed. See Table 4-1 on page 4-11 for complete details.

```
// JOB READ CARDS AND WRITE OUT TO TAPE
// ASSGN SYS001,491
// TLBL SORTOUT,'SORT.TAPE.OUTPUT'
// ASSGN SYS002,DISK,VOL=WORK01,SHR
// DLBL SORTWK1,'SORT.WORK.1',00/000,SD
// EXTENT ,WORK01,1,0,15,30
// EXEC SORT,SIZE=512K
  SORT FIELDS=(1,10,CH,A)
  RECORD TYPE=F,LENGTH=80
  INPFIL SYSIPT
  OUTFIL BLKSIZE=3200
  END
  ---- SORTIN SYSIPT data starts here ----
/*
/&
```

4.5.4 CAPRINT File Display Facility

The CAPRINT File Display Facility is composed of a multi-function exit that is entered during Phase 3 of a sort or merge. This exit (CAPRINT) will allow the output file of a sort/merge to be directed to SYSLST exclusively, or in addition to the normal SORTOUT file. CAPRINT includes three Phase 3 exits: E31 to read optional CAPRINT control statements, E32 to allow statement input to a merge when CAPRINT is active and E35 to actually perform the record display function.

CAPRINT provides the following capabilities:

- Any input file organized as sequential, ISAM or VSAM with either fixed- or variable-length records, may be displayed on SYSLST or directed to both SYSLST and SORTOUT.
- Using the INCLUDE/OMIT, OUTREC and SUM features of Unicenter CA-Sort, portions of a file can be selected for printing. The records printed may be listed in a number of formats (CHARACTER, HEX, and so forth) as desired.
- CAPRINT options may be selected as required for each execution, using either CAPRINT control statements or the DOS UPSI job control statement.
- Unless an alternate option is selected, each record in the file will be listed in both character and hexadecimal format.
- Using CAPRINT control statements, a title may be specified by the user to identify the display for documentation or control purposes.
- Input data to a MERGE may be read from SYSIPT using CAPRINT, providing a fast statement-to-print utility.

CAPRINT Options

There are two methods of specifying additional CAPRINT options or changing existing defaults:

- CAPRINT control statements
- DOS UPSI byte

Both provide adequate flexibility to allow changing options from step to step with a minimum of effort. The CAPRINT control statements are well suited for jobs run repeatedly or those job streams where the UPSI byte setting may affect other steps. The use of the UPSI job control statement is useful for individual debugging or testing runs.

4.5.5 CAPRINT Control Statements

- All control statements for CAPRINT are read from SYSIPT immediately following the Unicenter CA-Sort END statement.
- Each control statement must begin in position 2 of the statement. Only the first four characters of each keyword are significant.
- The E31 parameter must be specified on the Unicenter CA-Sort MODS control statement when control statements will be read from SYSIPT.
- Invalid CAPRINT control statements are ignored.

A summary of the CAPRINT Control Statements follows:

Control Statement	Function
DUMP	Lists each record in both character format and translated hexadecimal form, showing record lengths of each record along with a position spacer for reference. This is the default option.
CHARACTER	Same as the DUMP format; however, the record is only listed in character format. Unprintable characters are translated to blanks.
HEX	Similar to the DUMP option, except that each record is listed in only its hexadecimal representation. The zone portion of each character is printed on one line, with the numeric portion on the next.
LIST	Provides a more finished listing of each record in character format, without record lengths or position spacers.
TITLE	Places the 64 characters immediately following the keyword at the top of each new page of SYSLST output. This is useful for identifying printed files. If omitted, a default title will be inserted.
NOTITLE	Suppresses the printing of the title, date, and page number at the top of each page of SYSLST output.

4.5.6 Use of the UPSI Byte

If any UPSI setting is in effect for a sort/merge step using CAPRINT, it will override all options set using CAPRINT control statements (except TITLE).

Options may be selected using the following settings:

// UPSI 00000	=	DUMP
11000	=	DUMP
10000	=	CHARACTER
01000	=	HEX
00100	=	LIST
00001	=	NOTITLE

4.5.7 Dual Output File Capability

Normally, only a single output file, SORTOUT, may be produced during any sort/merge execution. However, CAPRINT provides the capability to write the output to SYSLST only (as a file display utility) or to direct a copy of the file to SYSLST as it is also being written to SORTOUT. This feature is useful for printing input transactions for verification and control purposes as they are being sorted, or listing data for a test file.

CAPRINT is able to automatically determine whether a SORTOUT file will be produced by examining the Unicenter CA-Sort OUTFIL control statement as follows:

OUTFIL EXIT assumes output is to SYSLST only.

OUTFIL BLKSIZE=nnn assumes output is to both SYSLST and SORTOUT.

4.5.8 Examples and Suggested Uses of CAPRINT

4.5.8.1 Example 1

Print a variable-length record, sequential file in character format. Only a specific record type will be listed.

```
// ASSGN SYS001,X'281'  
// TLBL SORTIN1  
// EXEC SORT  
  
MERGE FIELDS=COPY,ORDER=1  
RECORD TYPE=V,LENGTH=750  
MODS PH3=(CAPRINT,L5500,E31,E35)           Note 1.  
INPFIL BLKSIZE=7504  
OUTFIL EXIT                                Note 2.  
INCLUDE COND=(1,1,CH,EQ,C'A')  
END  
CHARACTER                                Note 3.  
/*  
/&
```

4.5.8.2 Example 2

Copy and list a VSAM KSDS file in hexadecimal format. Also title the output.

```
:  
:  
// DLBL SORTOUT,'VSAM.COPY',0,SD  
// EXTENT SYS001,DOSVS1,1,0,1,100  
// DLBL SORTIN1,'VSAM.FILE',,VSAM  
// EXTENT SYS002,DOSVS1  
// EXEC SORT,SIZE=100K  
MERGE FIELDS=COPY,ORDER=1  
RECORD TYPE=F,LENGTH=300  
INPFIL VSAM  
OUTFIL BLKSIZE=3000                        Note 4.  
MODS PH3=(CAPRINT,L5500,E31,E35)           Note 1.  
END  
TITLE   ***VSAM FILE***                    Note 5.  
HEX                                           Note 6.  
/*  
/&
```

Notes with Examples 1 and 2

1. Must be specified this way if CAPRINT Control Statements will be read.
2. Specifies SYSLST output only.
3. Suppresses hexadecimal printout.
4. Indicates a sequential SORTOUT file will be created in addition to the SYSLST print.
5. Overrides the default title.
6. Lists the file two lines per record in hexadecimal format.

4.5.8.3 Example 3

Display an ISAM file in both character and hexadecimal format. Since no additional options are desired, DUMP is assumed.

```
// DLBL SORTIN1,'ISAM,FILE',ISE
// EXTENT SYS001,DOSVS1,1,4,19
// EXTENT SYS001,DOSVS1,1,1,38
// LBLTYP NSD(03)
// EXEC SORT
SORT FIELDS=(1,4,BI,A),WORK=1
RECORD TYPE=F,LENGTH=95
INPFIL ISAM,BLKSIZE=95
OUTFIL EXIT
MODS PH3=(CAPRINT,L5500,E35)
END
/*
/&
```

Note 1.

Note 2.

4.5.8.4 Example 4

Load input transactions to disk and list them for verification.

```

:
// DLBL SORTOUT,'INVENTORY.TRANSACTIONS',0,SD
// EXTENT SYS001,DOSVS1,1,0,1,19
// EXEC SORT
MERGE  FIELDS=COPY,ORDER=1
RECORD TYPE=F,LENGTH=80
INPFIL EXIT
OUTFIL BLKSIZE=800
MODS   PH3=(CAPRINT,L5500,E31,E32,E35)
END
TITLE  INVENTORY TRANSACTIONS INPUT
LIST
/*
:
statement transactions go here
:
/*
/&
```

Note 3.

Note 4.

Without using exits, the control statements could be changed to:

```

MERGE  FIELDS=COPY,ORDER=1
RECORD TYPE=F,LENGTH=80
INPFIL SYSIPT
OUTFIL BLKSIZE=800,SYSLST,LIST,TITLE='INVENTORY TRANSACTIONS INPUT'
END
:
statement transactions go here
:
/*
```

Notes with Examples 3 and 4

1. MERGE FIELDS=COPY is not allowed for ISAM files. Sort the file on the ISAM key.
2. No control statement input therefore, only E35 is needed to display the file.
3. E32 is also included since the statement transactions will be read from SYSIPT into the MERGE after the CAPRINT control statements have been read by E31.
4. Produces a compressed character format listing of each record.

4.5.8.5 Example 5

Read statements into a sort and print them in character format.

```
// JOB CARDLIST USING NATIVE Unicenter CA-Sort UNIT RECORD FEATURES
// DLBL SORTWK1,'SORT.WORK.FILE',0,SD
// EXTENT SYS001,DOSVS1,1,0,1,19
// EXEC SORT
SORT   FIELDS=(1,5,CH,A),WORK=1
RECORD TYPE=F,LENGTH=(80,,80)
INPFIL SYSIPT
OUTFIL SYSLST,CHAR,TITLE='***** CARD FILE LISTING *****'
END
:
statement input data goes here
:
/*
/ &
```

4.5.9 Unicenter CA-Sort and Advantage CA-Easytrieve PLUS Interface

The Advantage CA-Easytrieve PLUS reporting system may be used as an E35 exit for report generation in addition to the CADART and CAPRINT facilities.

The Unicenter CA-Sort and Advantage CA-Easytrieve PLUS interface allows sorted output to be sent to and processed directly by Advantage CA-Easytrieve PLUS. This eliminates the associated overhead of an external storage media.

For more information on Advantage CA-Easytrieve PLUS reporting facility, refer to the related publications.

4.5.9.1 Examples and Suggested Use of Advantage CA-Easytrieve PLUS

Step 1. Compile and link edit/catalog the Advantage CA-Easytrieve PLUS program.

```
// EXEC EZTPA00

      PARM LINK(EZTPMOD) 1

      FILE FILEA F 80 EXIT (EZTPSORT USING(PARM-REGISTER)) 2

      ....
      ....
/*
// EXEC LNKEDT
/*
```

Step 2. Run the sort job.

```
..

// ASSGN SYS001,cuu
// DLBL SORTIN
// EXTENT SYS001
// EXEC SORT
  SORT FIELDS=(1,9,CH,A)
  RECORD TYPE=F,LENGTH=80
  INPFIL BLKSIZE=800
  OUTFIL EXIT
  MODS PH3=(CAEZTP,L1500000,,E35) 3
  END
/*
/&
```

4.5.9.2 Notes

- 1 The default cataloged phase name is EZTPMOD (the name of the phase that the CAEZTP E35 exit will look for.) This phase name may be overwritten by specifying a different SORTOUT file name in the FILNM operand of the OPTIONS control statement. The phase name must start with **EZTP**.
- 2 EZTPSORT is the name of the Advantage CA-Easytrieve PLUS and Unicenter CA-Sort interface and is supplied with the product. The USING parameter must be specified as indicated.
- 3 CAEZTP is the name of the Advantage CA-Easytrieve PLUS E35 exit and is supplied with the product. The length should be specified as the size of the EZTPMOD program plus 4K.

Glossary

access method. A technique for moving data between main storage and input/output devices. CA-SRAM operates as an access method. The programmer controls the writing and reading of data to temporary disk storage using OPEN, READ, WRITE and CLOSE commands.

archive. Copies data onto tape and deletes them from disks. Archiving is one method of increasing available space on Direct Access Storage Devices.

automatic in-core sort. CA-SORT attempts to perform every sort entirely within main storage. You do not need to provide intermediate work files or change existing JCL statements to invoke the facility. The work files are opened only if an in-core sort cannot be performed.

automatic secondary allocation. CA-SORT automatically allocates secondary extents on your disk work and output data sets if there are insufficient primary extents.

bias. An integer value used to indicate the "sortedness" of the input records processed.

block. Any kind of grouped data (a string of records, a string of words, a character string) treated as a single unit.

CA-SRAM. Sort Reentrant Access Method. Invokes the sort from high level languages such as COBOL, FORTRAN, and PL/I. Input records are written to the CA-SRAM work files as sorted strings and retrieved in sequence by standard subroutine calls. CA-SRAM operates like an access method, reading and writing sorted data to and from its disk work file. CA-SRAM allows the activation of several sorts concurrently.

catalog. File the computer system maintains as a directory index to the systems' files, similar to the card catalog of a standard library.

control block. A storage area that a program uses to hold control information.

control field. A portion of records you designate, which are used by the program to sort or merge records into a specific sequence.

CSECT. Control section.

CSL. Condensed Source Listing. VSAM file which holds the COBOL program source information.

DASD. Direct access storage device. Any storage device that can immediately read or write anywhere on its recording surface. Often made up of a number of rotating disks mounted on a shaft accessed by arms containing read/write heads, and working on a principle similar to that of a phonograph. Data is written magnetically on each disk, and given an address. It may then be accessed either sequentially or randomly at a very high speed.

default option. An option which has already been set for the system according to the way it is assumed to be most generally used. You may change any default option to your own specifications.

direct access dataset. A data set whose records are stored or retrieved according to their actual addresses or addresses relative to the beginning of the data set.

DSN. Data set name. A term or phrase that is the entire, fully-qualified name of a file.

exit. A program instruction which may be used to escape one set of instructions and pass control to another.

fixed length records. These records have the same length as all other records with which they are associated.

installation option. A value specified on the installation macro when the product is installed. This value controls scheduling operations throughout the data center unless it is overridden at the schedule or job level.

integrated file utility. Allows you to input files with certain attributes and output those files with the same or different attributes. Allows you to copy, select, and reformat records.

ISAM. Indexed sequential access method. Allows for both sequential and random access to data stored on disks by identifying each record through a key. It then uses track, cylinder, and master indexes to locate that record within a file.

job. A job stream consisting of one or more DOS/VSE jobs.

keyword. A part of a command that consists of a specific character string.

LIOCS. Logical Input/Output Control System.

load module library. A partitioned data set which contains load modules that you need to store for later retrieval.

main storage sort. Small data sets that can be contained within main storage can be sorted without any work data sets.

modal length. The record length that occurs most frequently in the input file to be processed.

module. A discrete program unit which may be compiled, loaded, and combined with other units.

multi-volume. Indicates that a data set spans more than one tape volume.

PDS. Partitioned data set. A data set in direct access storage that is divided into a directory and partitions called members. Each member can contain a program, part of a program, or data. Another name for a program library.

record. A group of logically related fields treated as a unit.

reentrant. The attribute of a program or routine that allows the same copy of the program or routine to be used concurrently by two or more tasks.

return code. Information that is returned to the calling program from the program or routine it called to perform a particular function. The information that is returned will then be used by the calling program.

sort pass. During the execution of a sort program, a single processing of all items of a set for the purpose of reducing the number of strings of items and increasing the number of items per string.

string. An ordered list in control field sequence.

user exits. Computer instructions that allow you to modify the software package's code to meet your data center's needs.

utility functions. Run general-purpose programs that perform some activity not specific to any other particular application.

variable length records. Records that are not uniform in length.

VOLSER. Volume serial number. An alphabetic and/or numeric code assigned to each tape or disk volume.

volume. A reel of magnetic tape you mount and dismount as a unit for storing a collection of information.

VSAM. Virtual Storage Access Method. Allows the transfer of information between the CPU's main memory and a direct access storage device. Records organized in logical key field sequence, in physical creation, or by relative-record number, can be accessed directly or sequentially.

VTOC. Volume Table of Contents. Lists all the files, with their addresses, that a disk pack contains. Also called a directory.

Index

Special Characters

\$FIINTR1 module 4-54

A

Abnormal conditions 2-61
AC 2-7, 2-11
Address constants 3-5, 3-7
ADDROUT
 keyword 2-10, 2-53, 3-15
 with VSAM 2-25, 2-53
Advantage CA-Easytrieve PLUS
 and Unicenter CA-Sort interface 4-66
 using 4-66
Allocating storage 2-56
Alternate collating sequence 2-49, 4-45
ANALYZE control statement 2-9, 2-63
AND 2-36
AQ 2-7, 2-11, 2-49
AQTT table 4-45
ASCII 2-9, 2-29, 2-33
ASSGN statement 4-1
Auditing 4-16

B

BI 2-7, 2-8, 2-11
BIAS 4-18
Bit control fields 2-17
BLKSIZE keyword 2-8, 2-29, 2-32, 2-34
Block
 count 3-11, 3-16, 3-19, 3-25
 descriptor word 2-24
Branch table for exits 3-5, 3-7
BUFOFF keyword 2-30, 2-33
BYPASS keyword 2-29, 2-33, 3-17

C

CA-SORT
 keyword examples 4-22
CADART keyword 2-8, 2-33, 2-34, 4-66
CAIE15 exit 4-57, 4-58
CALC keyword 2-9, 2-63
CALCAREA 2-10, 2-64

Calling sequence 3-19, 3-24
CAPRINT
 capabilities 4-59
 control statements 4-60, 4-66
CASCADE
 keyword 2-10, 2-20, 2-54
 sorting technique 2-20
CCB address 3-16, 3-25, 3-26
CH 2-8, 2-9, 2-11, 2-18
CHALT keyword 2-54
CHAR keyword 2-8, 2-33
CHECK keyword 2-10, 2-54
Checkpoint 2-14, 4-4
CHKPT|CKPT keyword 2-7, 2-13
CISIZE
 for input and output 4-2, 4-3
 for VSAM/SAM work files 4-9
 maximum 1-11
CKD DTF 4-6
CLO 2-7, 2-11
CLOSE keyword 2-8, 2-30, 2-33
Closing files 3-11, 3-16, 3-19, 3-25
COBOL
 initiating Unicenter CA-Sort 4-48
 SORT verb 4-44
CODE keyword 2-9, 2-49
Collating sequence 2-49, 3-19, 4-45
Comments 2-4
Compatibility with other sorts 2-25, 2-29, 2-33
COND keyword 2-9, 2-34
Continuation 2-4, 2-5
Control
 field
 bit 2-17
 considerations with SUM 2-48
 length 2-10
 number of 2-14
 parameters 2-14
 position 2-10
 relationship to INREC/OUTREC 2-38
 with ADDROUT 2-53
 statement
 format 2-2, 2-6
 types 2-6
COPY utility 1-8, 2-17

CSL 2-7, 2-11
CST 2-7, 2-11
CTO 2-7, 2-11

D

DATA keyword 2-8, 2-30
DEFDTF, installation parameter 4-6
DELBLANK keyword 2-8, 2-17, 2-23, 2-37
Deleting records 2-34, 2-47, 3-15, 3-19, 3-23, 3-31
Device assignment defaults 4-11
Devices
 input 1-4
 output 1-4
DIAG
 keyword 2-10
 OPTION keyword 2-54
 SYSPARM parameter 2-61
Diagnostics 2-10, 2-61
Disk space management systems 4-2, 4-5, 4-51
DISK, installation parameter 1-11, 2-63
DISP 4-9
DLBL statement 4-1, 4-3
DOS/VS considerations 2-34
DSPSIZE
 OPTION keyword 2-54
DTF determination 2-61, 4-6
DUMP
 keyword 4-60
 OPTION keyword 2-10, 2-54
 OUTFIL keyword 2-8, 2-33
Duplicate records 2-47, 2-48, 3-31
DYNAMD keyword 2-54
DYNAMFI
 installation parameter 4-54
 keyword 2-10, 2-54, 4-54

E

END control statement 2-9, 2-65
Entry point 3-5
EODAD exit 3-28
EQ 2-36
EQUALS keyword 2-7, 2-10, 2-13, 2-14, 2-55
ERASE keyword 2-55
ESDS 2-9, 2-30, 2-32, 2-34
Execution
 Phase 0 - Initialization 1-2
 Phase 1 - Sort 1-2
 Phase 2 - Intermediate Merge 1-3
 Phase 3 - Final Merge 1-4

Exit E11 3-9, 3-11
Exit E15 2-23, 2-29, 2-30, 2-38, 2-48, 3-9, 3-14
Exit E17 3-9, 3-16
Exit E18 3-9, 3-16, 3-26
Exit E21 3-9, 3-17
Exit E25 3-9, 3-19
Exit E27 3-9, 3-19
Exit E31 3-9, 3-19
Exit E32 3-21
Exit E35 2-23, 2-32, 2-34, 2-38, 2-48, 3-9, 3-23
Exit E37 3-9, 3-25
Exit E38 3-9, 3-25, 3-26
Exit E39 3-26
EXIT keyword 2-8, 2-9, 2-29, 2-32, 3-21, 3-23, 3-24
Exit routines
 changing record length 1-11
 how to specify 2-26
 length of 2-26, 2-28
 loading of 3-5
 parameter summary 3-9
 preloaded 2-26, 2-28, 3-5
 sample coding 3-31
 self-relocating 2-28
 storage management considerations 2-28
 usage with other control statements 3-3
 uses for 3-8, 3-10
Explicitly defined VSAM/SAM files 4-8
EXTENT statement 4-1, 4-3, 4-5

F

FBA DTF 4-6
FI 2-8, 2-11, 2-34
FIELDS keyword 2-8, 2-9, 2-18, 2-38, 2-47
File
 DA type 4-3
 names 2-10, 2-61, 4-1
 organizations
 MERGE specifications 1-8, 2-14
 SORT specifications 1-7, 2-14
 positioning options 2-32
FILES keyword 2-13, 2-19
FILNM keyword 2-10, 2-22, 2-55
Fixed-length records
 MERGE specifications 1-8
 OPTION control statement 2-53
 RECORD control statement 2-24
 SORT specifications 1-6
 SORTx control statement 2-23

FL 2-8, 2-11
FORMAT keyword 2-13, 2-14, 2-18, 2-36, 2-47
FREEOUT keyword 2-9, 2-33

G

GE 2-36
General register contents 3-7, 3-8, 4-43
GT 2-36

H

Hardware requirements
 maximum 1-6
 minimum 1-6
HEX keyword 2-8, 2-33, 4-60

I

I/O error processing 3-16, 3-25, 3-26
IDCAMS 4-8, 4-9
Implicitly defined VSAM/SAM files 4-2, 4-8
In-core
 sort 2-64, 4-4
 SORT specifications 1-7
INCLUDE control statement 2-9, 2-34
INCLUDEx control statement 2-9
Initiation
 by JCL 1-2, 4-1
 from another program 1-2, 4-43, 4-47, 4-49
INPFIL control statement 2-8, 2-29, 4-8
INPUT keyword 2-8, 2-30
INREC control statement 2-9, 2-38
Inserting records 2-30, 3-14, 3-15, 3-24
Installation
 macro 1-11, 4-6
 parameters 1-11
Interface
 Unicenter CA-Dynam/D 4-51
 Unicenter CA-Dynam/FI 4-54
Intermediate storage 1-7
Invoked
 operation 1-2
 parameter list 4-43
ISAM
 files
 MERGE specifications 1-8
 SORT specifications 1-7
 keyword 2-8, 2-29

J

JOB statement 4-1

K

KEY
 keyword 2-7, 2-13, 4-15
 symbolic names 4-15
KEYLEN keyword 2-55
KSDS 2-8, 2-32, 2-34

L

LABEL keyword 2-10, 2-55, 4-6
Labels
 header 3-11, 3-19, 3-28
 non-standard 2-61
 nonstandard 3-11, 3-16, 3-19, 3-25
 standard 4-1
 trailer 3-11, 3-16, 3-19, 3-25, 3-28
 use in DTF determination 2-61, 4-6, 4-7
LE 2-36
LENGTH keyword 2-7, 2-23, 2-24
Lengthening records 2-38
LERAD exit 3-28
Linkage conventions for user exits 3-7
LIOCS 2-34
LIST keyword 2-9, 2-33, 4-60
LOAD macro 3-5, 4-43
Logical unit
 default 4-11
 default file names 4-1, 4-3
 for user label processing 3-19
 overriding 2-19, 2-61
 use in DTF determination 2-61, 4-6
 user label processing 3-11
 work file 2-61
LT 2-36

M

Main storage, requirements 1-6, 1-11
MERGE
 considerations with INREC 2-41
 control statement 2-7, 2-13, 2-20
 COPY function 1-8, 2-13, 2-17
Merge order 1-4
Messages
 printing of 2-9, 2-61
 routing of 2-9

Modal record length 2-23
MODS control statement 2-8, 2-26, 2-28
Multiple output files 2-19, 2-20, 4-14

N

NE 2-36
NOCI keyword 2-55
NOCIFORMAT 4-8
NODUMP keyword 2-10, 2-55
NOEQUALS keyword 2-10, 2-13, 2-22, 2-55
NOINC keyword 2-10, 2-55, 4-3
NOMIX keyword 2-10, 2-55
NORC keyword 2-55
NOTITLE 4-60
NOTPMK keyword 2-10, 2-32, 2-56
NOVERIFY keyword 2-9, 2-56
NRECS keyword 2-10, 2-34, 2-56, 4-16
Null files 1-6

O

OMIT control statement 2-9, 2-34
OMITx control statement 2-9
OPEN keyword 2-8, 2-9, 2-29, 2-32
Opening files 3-11, 3-19
Operation type
 ALTSEQ 2-3
 ANALYZE 2-3
 END 2-3
 INCLUDE 2-3
 INPFIL 2-3
 INREC 2-3
 MERGE 2-3
 MODS 2-3
 OMIT 2-3
 OPTION 2-3
 OUTFIL 2-3
 OUTREC 2-3
 RECORD 2-3
 SORT 2-3
 SORT1-SORT9 2-3
 SUM 2-3
OPT keyword 2-56, 2-61
OPTION control statement 2-10, 2-52
OR 2-36
ORDER keyword 2-8, 2-13
OUTFIL control statement 2-8, 2-32, 4-8
OUTPUT keyword 2-9, 2-33
OUTREC control statement 2-9, 2-22, 2-24, 2-38

Overriding
 installation options 2-52
 logical unit assignments 2-61

P

Page fixing 2-56
Parallel sorting technique 2-20
Parameters
 conventions
 for invoked operation 4-43
 for user exits 3-7
 list 4-44
 passed to user routines 3-9
PARAMS keyword 2-10, 2-56
Passwords 3-27
PCHLST 2-33, 4-57
PD 2-8, 2-11, 2-18, 2-34
PD0 2-11
Performance 2-20, 2-61
Pooling 4-7
PREOPEN keyword 2-10, 2-56
PRESEQ keyword 2-8, 2-30
PRINT keyword 2-10, 2-56
Program
 capabilities 1-6
 organization 1-2
PUNCH 2-33, 4-57

R

RDW 2-24, 2-38
Record
 descriptor word 2-24
 length
 logical 1-10, 1-11
 maximum 1-10, 1-11
 MERGE specifications 1-8
 of input file 2-23, 2-24
 SORT specifications 1-6
 vector 3-14, 3-21, 3-23
 selection 2-17, 2-34, 2-37
 summarization 2-47
RECORD control statement 1-11, 2-8, 2-23
RECORDFORMAT 4-8
RECORDS parameter 4-8
Reformatting records 2-38, 2-41
Restart 4-4
RESTART keyword 2-10
Return
 address 3-8

Return (*continued*)
 codes with exits 2-30, 3-14, 3-15, 3-17, 3-21, 3-24
 REUSE keyword 2-32, 4-9
 Rotational Positional Sensing 1-4
 ROUTE keyword 2-10, 2-57
 RPS 1-4
 RRDS 2-9, 2-32, 2-34

S

SAM files 2-53
 Save area 3-8
 Secondary allocation, work files 1-6, 4-3, 4-5, 4-9
 Shortening records 2-38
 SIZE
 file estimate 2-63
 keyword 2-8, 2-13
 on EXEC statement 2-30, 2-34, 4-1
 SKIPBYTE keyword 2-8, 2-30
 SKIPREC keyword 2-10, 2-34, 2-57, 4-16
 SORT control statement 2-7, 2-10, 2-20
 SORT/MERGE operation 4-1
 SORTCKP 4-4
 SORTIN keyword 2-10, 2-57
 SORTOUT keyword 2-10, 2-57, 4-1
 SORTWK keyword 2-10, 2-57
 SORTx
 control statement 2-8, 2-18
 example 4-14
 specifications 2-21
 SPAN keyword 2-9, 2-25, 2-34
 Spanned records
 MERGE specifications 1-8
 SORT specifications 1-6
 Specifications
 MERGE 1-8
 SORT 1-6
 Sort/Merge 1-6
 Statement definer 2-10
 SUM
 control statement 2-9, 2-22, 2-47
 usage with SORTx 2-48
 Symbolic unit names 4-11
 SYSIPT
 keyword 2-8, 2-30, 2-65, 4-57, 4-58
 logical unit 1-6
 use of exits with 2-30
 SYSLOG logical unit 1-6
 SYSLST
 keyword 1-6, 2-33, 2-34, 4-57
 logical unit 1-6

SYSLST (*continued*)
 use of for sort output 2-33
 SYSNO keyword 2-8, 2-10, 2-19, 2-57
 SYSPARM 2-61
 SYSPCH, use of for sort output 2-33

T

Tables
 Parameters Passed at Sort/Merge Exits 3-9
 Sort/Merge Device Defaults 4-11
 Supported Devices 1-5
 Table Maximum LRCL 1-10
 Table Summary of Control Statements 2-7
 Uses for Sort/Merge Exits 3-10
 Tag sort 2-58, 4-18
 Tape
 management systems 2-61
 unlabeled 4-6
 TITLE keyword 2-8, 2-33, 4-60
 TLBL statement 4-1
 TOL keyword 2-8, 2-30, 2-32, 2-34
 TP keyword 2-10, 2-57
 TYPE keyword 2-23

U

Unicenter CA-Dynam/D interface 4-5, 4-51
 Unicenter CA-Dynam/FI interface 2-13, 4-54
 Unicenter CA-Sort
 and Advantage CA-Easytrieve PLUS interface 4-66
 macro 4-6, 4-54
 Unicenter CA-SRAM 2-20
 Unit record
 input 2-29, 4-58
 support 4-57
 Unlabeled tape 2-32
 UPSI byte 4-61
 User Exits 3-1, 3-10, 3-11

V

Value 2-3
 Variable-length records
 INCLUDEx/OMITx control statement 2-38
 MERGE specifications 1-8
 OUTFIL control statement 2-34
 RECORD control statement 2-23, 2-24
 SORT specifications 1-6, 1-8
 VERIFY keyword 2-9, 2-57

VOLUME keyword 2-8, 2-29

VS 2-34

VSAM

exit lists 3-26, 3-28

files

Exits E18 E38 E39 3-27

INPFIL control statement 2-29, 2-30

MERGE specifications 1-8

OPTION control statement 2-53

OUTFIL control statement 2-32, 2-34

SAM data sets 4-8

SORT specifications 1-6

using optional keywords 2-25

keyword 2-8, 2-29, 4-8

passwords 3-26

VSAM/SAM

explicit define 4-8, 4-9, 4-15

files 2-34, 4-2, 4-7, 4-8, 4-13

implicit define 4-2, 4-8, 4-9, 4-13

input files 2-29, 2-30

support 1-7

work files 4-5, 4-8, 4-9

Z

ZD 2-8, 2-9, 2-11, 2-36

W

WKASN keyword 2-58, 2-61

Word 3-14, 3-21, 3-25, 3-28

Work file

allocation 4-3

assignment 2-58

secondary allocation 4-3, 4-5

SORT specifications 1-7

space calculation 2-63, 4-4

specification 4-4

tape 1-6

types 4-3

WORK keyword 2-8, 2-11

WORKNM keyword 2-10, 2-22, 2-58

Wrong-length record 3-17

Y

Y2B 2-11

Y2C 2-11

Y2D 2-11

Y2P 2-11

Y2S 2-11

Y2Z 2-11

Year 2000 keyword examples 4-22

