SB11-5351-0

Conversion Aid System/3 to DOS/VSE
Program Description and Operations Manual

Program Number 5785-KAD

International Field Program

PREFACE

This publication provides the information required by a user who want to migrate from a System/3 to IBM 4300 Processor under DOS/VSE. It contains the information needed to convert programs and files and to prepare the corresponding control card. The assistance concerning file distribution is more meaningful when the user is familiar with the physical location of these files.

The publication is divided into three main parts: "Understanding the Program" contains a functional description of the program and presents hardware and software requirements.

"Installing the Program" describes the components of the program as distributed and how to install them.

"Running the Program" contains the steps in performing the conversion of a job stream and describes the input and output.

Appendix A and Appendix B contain a detailed description of each component of the program and of the layout of the files used by these components. These appendixes should be consulted for modifications to the source code. Appendix C contains a sample problem, illustrating the conversion of the job streams of two programs.

All listings are in English, and all constants are in tables o: arrays at the end of each program. This makes it easy to modify them or to tranlate them into another language.

In order to use the System/3 to DOS/VSE Conversion Aid program, the reader should be familiar with the differences between System/3 RPG and DOS/VS RPG. He must also be familiar with the JCL, sort specifications and programs, and the charateristics of the different access methods.

Before using this International Field Program, the reader should be familiar with the contents of *System/3 to DOS/VS Conversion Guide,* GC20-1792 and *Introduction to DOS/VS,* GC33-5370.

First Edition (July 1979)
This edition applies to Version 1, Modification Level 0, of Conversion Aid System/3 to DOS/VSE, Program Number 5785-KAD, and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are periodically made to the information herein; before using this publication in connection with the operating of IBM systems, consult the latest *IBM System/370 Bibliography,* GC20-0001, for the editions that are applicable and current.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to:

IBM Netherlands
International Field Program Center
P. O. Box 24
1420 AA UITHOORN
Netherlands

When a System/3 user decides to upgrade to a System/370 or an IBM 4300 Processor, he encounters a certain number of difficulties. First, System/3 RPG, in which most user programs are written, does not have the same functions as DOS/VS RPG. Second, within a short time the user has to change not only his programs, but also the control cards and physical location of the files.

IBM has made the System/3 to DOS/VSE Conversion Aid International Field Program available in order to help the System/3 user convert to DOS/VSE and allow him to prepare for the conversion using his System/3. In this way, while using tools familiar to him, he can have his programs ready for execution when his System/370 or 4300 is installed.

It is recommended that System/3 sequential files without additions to the end of the file and treated consecutively be converted to SAM files. Otherwise files should be converted to VSAM files.

The System/3 to DOS/VSE Conversion Aid International Field Program consists of 19 source programs written in System/3 RPG and one resident program written in DOS/VSE Assembler language. The DOS/VSE program should be catalogued into the DOS/VSE relocatable library. This routine serve as a complement to the converted System/3 programs when the chosen methods is VSAM, and it is used to access on ESDS VSAM file or to access the Data component of a KSDS VSAM file by relative record number.

THE CUSTOMIZER

The first four programs to be executed make up the customizer; two of these programs can be executed on a System/3 Model 10 and the other two on a System/3 Model 15-B. The following is specified:

- Using the SWITCH card:
  - Control statement input on 80- or 96-column cards or on diskette
  - Customized output of the conversion aid programs on 80- or 96-column cards, diskette, or magnetic tape
  - Output of the components to be cataloged on the DOS/VSE relocatable library on 80-column cards, diskette, or magnetic tape.
- Configuration of the System/3 on which the conversion programs are to be executed:
  - Input of the programs to be converted to 80- or 96-column cards or diskette
  - Type of disk unit for work files
  - Output of the converted programs (including control cards for cataloging and printer carriage control tapes) on 80- or 96-column cards, diskette, or magnetic tape
  - Output of the input job stream for execution under DOS/VSE on 80- or 96-column cards, diskette, or magnetic tape.
- Correspondence between System/3 devices and System/370 devices
- Description of System/370 or 4300 configuration:
  - Type of disk (3330, 3340 or FBA)
  - Type of printer
  - Printer line length (default value)
  - Printer overflow line (default value)
  - System/370 POWER/VS (included or excluded)

This customizer will call an input file that contains the remaining components of the conversion aid program.

If this customizer needs to be executed on a System/3 Model 15-C or 15-D, programs CST151 and CST152, residents in Source Library, must be compiled. The SUBR93 subroutine, needed to compile CST151 program, is cataloged in R Library.

THE CONVERSION PROGRAM

CONV00 and CONV01, the first two components of the actual conversion program, prepare the definition and calculate the occupancy of all the files of the programs to be converted.

The CONV00 control cards specify the characteristics of all files: the type and name of each file, the length and number of records, and so on. CONV01 provides a listing of file occupancy of the logical units, and places on a work file the information required by subsequent components.

Sequential disk files can be converted to SAM or VSAM-ESDS. Direct files can be converted to VSAM-RRDS with relative record number.

Indexed files can be converted to VSAM-KSDS or VSAM-ESDS with alternate indexes (if the file is created unordered).

The CONV02 component reads the System/3 source programs. If these programs are compatible with the file definitions, and if there are no other unresolvable incompatibilities, CONV02 places them in a work file. CONV02 is the only component that is executed each time a source program is converted. If a program is not accepted, only that program need be resubmitted.

It is assumed that all programs to be converted have been executed without error on the System/3. The CONV03 to CONV05 components convert all the source programs in a job stream placed in the work file by CONV02. CONV05 places the converted source programs on the type of media specified in the customizer.

The listings produced for each converted program contain:

- A listing of the converted program
- Modifications performed by the conversion aid program
- Modifications assumed by default and which must be verified
- Functions that the program cannot convert and which must be converted manually.

The conversion aid program contains a large number of components in order to execute it on a System/3 16K partition.

The programs converted using components CONV00 to CONV05 can be executed under DOS/VSE. The remaining components perform the following functions:

- Creating printer carriage control tapes
- Distributing files on disk
- Preparing the conversion of System/3 files to System/370 via intermediate tape files or Data Import Feature
- Defining catalog, space, and VSAM files
- Preparing the input job stream for execution.

CREATING PRINTER CARRIAGE CONTROL TAPES

The CONV06 component processes printer carriage control tapes. First, it reduces their number to a minimum; then it numbers them, starting with the number specified in the input to the CONV04 component. In addition, it gives a name to each tape and lists its format. The component generates the control and data statements used to compile and catalog all the FCBs, unless an IBM 1403 Printer without POWER/VS is to be used.

DISTRIBUTING FILES ON DISK

The CONV07 component distributes files on disk. A data statement must be entered to CONV07 for each SYSXXX defined in CONV00 to relate the SYSXXX to the volume number and to indicate the relative position in which the disk space of the pack must be allocated. If these statements contain no errors, CONV07 produces an occupancy list similar to the one produced by CONV01. This list indicates the files on each pack which may contain one or several SYSXXX. CONV07 produces, too, a listing of the DLBLs and TLBLs in alphabetical order according to labels and with blank file names.

DOCUMENTATING INSTALLATION AND FILES CONVERSION

Components CONV08 to CONV10 document installation and prepare the conversion of files from System/3 to System/370 or IBM/4300. Files can be converted via tape medium (concerns only those files whose definition statement contain an R in column 65) or via Data Import utility if 3340 Data Import Feature is installed. (Concerns only those files whose definition statement contain U in column 65.)

The CONC09 component lists all control statements required to execute the programs converted at this point. This listing is in alphabetical order according to program name. The control statements are prepared assuming the pograms use the NAME-LABEL relationship specified in CONV02. These listings may be useful if the components that prepares the input job stream for execution are not used. Again, the CONV09 component prepares if any file definition statement was specified with an R in column 65:

- The $TINIT statements to initalize tapes
- The $COPY statements needed to dump tape files with a block size no larger than 4096 bytes.

The CONV10 component prepares:

- DEFINE MASTER CATALOG statements, to which must be added the number of the first track/block of the EXTENT and the name of the disk pack from the EXTENT statement and from the VOLUME parameter in the corresponding AMS statement.
- DEFINE SPACE statements.
- DEFINE CLUSTER statements to every VSAM file defined.
- DEFINE ALTERNATE INDEX and its PATH to every VSAM file defined with type A specified.
- REPRO statements (concerns only to those files whose definition statement contains an R in column 65).

- DATA IMPORT UTILITY statements (concerns only to those files whose definition statement contains a U in column 65).

- BUILD INDEX statements (concerns only those files whose definition statement contains an R or an U in column 65 and type A specified).

PREPARING THE INPUT JOB STREAM FOR EXECUTION

Components CONV11 to CONV14 convert System/3 OCL to DOS/VSE JCL. Input can be any set of System/3 OCL control statements, but /*, /&, and /. records must be replaced manually by */, &/, and ./ in order to avoid terminating the program on an end-of-file condition.

If an OCL stream is unexpectedly terminated, the CONV11 component can be executed again, beginning with the data that was not processed. The created records will be added to the output file.

If the OCL statements contain no errors, and if all programs and files referred to in the statements have already been converted:

- The DOS/VSE JCL statements required to execute the converted programs are generated.

- System/3 COPY statements are converted to AMS REPRO statements.

- System/3 SORT control statements are converted, with the exceptions given in Appendix A.

  Special care is required when converting SORT control statements using the digit-portion option. These statements are not converted, unless the numeric field is changed from a D to a C or a U.

The remaining OCL statements are transferred without change, and a warning message is issued.

ADDITIONAL CONVERSION MODULES

When a System/3 program processes randomly a sequential file converted to ESDS, the converted program calls the WKVMXV module.

PROGRAM CONVERSION FUNCTIONS

The following functions are performed by the components of the program:

- The H statement is transferred with its valid entries.

- All statements are numbered and contain the name of the program.

- The required entries of the F statement are changed according to the file characteristics and the logical SYSXXX unit.

- File names are decreased to seven characters and are modified (if required) to avoid duplicates. The corresponding F, E, I, C, and O statements and file conversion cards are updated.

- The first ten compilation tables are modified to fit into 80-column cards (if they exceed 80 positions).

- The E statement is modified from execution tables or arrays if the number of entries per record is to be modified. The number of entries per record is also modified in the E statement if the record length has been changed.

- New table files are created when more than one execution table is assigned to a file.

- The L statement is converted to comments, or a new L comment statement is created for a printer file.

- Decimal positions are placed in arrays or in elements of arrays defined in I statements.

- Half-adjust (H in position 53) is added for SQRT operations.

- Module to convert relative record number to relative byte address (for VSAM ESDS files) is included and flagged.

- C statements added to branch to the included module are updated.

- C statements are added to create random RRDS files.

- The ADD characters in the addition to sequential files are suppressed.

- If nothing is added to an additions file, this condition is detected.

- Errors in operation codes are detected.

- In O statements, the skip to line number by channel SKIP is changed.

- One space is assumed following printer lines without spacing or SKIPs.

- The overflow indicator is changed from output AND lines to primary lines.

- The L0 indicator in C statements is suppressed.

- The specification of output spacing in console files is suppressed.

MODIFICATIONS TO BE VERIFIED

The following conversion actions performed by the program may not be complete. They should be verified before compiling the converted source programs.

- A warning message is issued to remind the user that an AMS Build Alternate Index Command must be executed after running a program in which an ESDS file with alternate index is created.

- The stacker selection specified in I and O statements is ignored.

- Files processed between limits and random, primary, display, demand, or table files are assumed not to be affected by an external indicator.

- The record length in ADDROUT files is changed.

- A warning message is issued to indicate that the contents of the *ERROR field should be analyzed if M fields, record sequence checking, or record identification checking are used in the program.

FUNCTIONS NOT CONVERTED

The following functions are not converted. Either the conversion must be performed manually, or the System/3 source programs must be modified before conversion.

- Table expansion is performed only for the first ten defined compilation tables. The eleventh table and those following must be modified manually if the data does not fit into 80-column statements.
- A warning message is issued to indicate that the execution tables must be modified if the number of entries per record has been changed.
- A warning message is issued when data files are converted from 96-column to 80-column cards, and the defined fields exceed the new record length.
- Combined files are not allowed.
- Double I/O areas and blocking of SPECIAL files are not allowed.
- A warning message is issued when H indicators are used.
- The module used by SPECIAL files or by any files called by an exit routine must be rewritten.
- Statements added using an Auto Report COPY statement must be cataloged in the R sublibrary of the DOS/VS source statement library.
- Only 15 files are allowed per program.
- Only one printer is allowed per program.
- No console files, other than output or display files, are accepted.
- No more than 11 skip positions are allowed in a printer listing.
- Data cannot be printed on cards.
- Warning messages are issued to indicate reserved words that must be changed.

MISCELLANEOUS CONVERSION AIDS

- A customizer adapts the programs to be converted to the System/3, System/370 or 4300 configurations used.
- Printer carriage control tapes are converted, and the cataloging of FCBs is prepared.
- Files and data spaces are distributed on disk.
- The execution statements of each program are listed.

- Control statements to convert files to disk using magnetic tape as intermediate step are generated.

- Execution job streams, including about 80% of sorts and file copies, are converted from System/3 to System/370 or 4300.

MINIMUM CONFIGURATIONS

The minimum System/3, System/370 or 4300 configurations required by the System/3 to DOS/VSE Conversion Aid International Field Program are given below.

SYSTEM/3

The customizer components of the program require, in addition to the configuration needed to execute the source programs to be converted (System/3 Model 10 or 15), one IBM 5444 Disk Storage Drive or one IBM 3340 Direct Access Storage Facility, and one magnetic tape unit.

The program conversion components require:

- One System/3 Model 8, 10, 12, or 15
- A minimum of 16K bytes of partition core storage
- One data entry unit, any type
- One 80- or 96-column card punch, or one magnetic tape unit
- One printer, any type
- One disk unit, any type.

The System/3 must be equipped with a DSM operating system and an RPG compiler.

SYSTEM/370 OR 4300 PROCESSOR

To catalog the converted source programs, any System/370 or 4300 can be used, equipped with:

- One 80- or 96-column card reader, one magnetic tape unit, or one Diskette Input/Output Unit

- One printer

- One IBM 3330 Disk Storage or one IBM 3340 Direct Access Storage Facility or one IBM 3310 or IBM 3370.

In order to convert System/3 files using the conversion program, a tape unit or Data Import Feature is also required.

The characteristics of the System/370 or 4300 must be those specified to the customizer components. The system must include work areas with cataloged labels in order to compile and execute the sort programs and the BUILD ALTERNATE INDEX command.

IBM DOS/VSE RPGII, Licensed Program 5746-RG1 is required. To convert System/3 files using the control statements generated by the conversion program, VSAM support is required.

An overall flowchart of the program is given in Figure 1.

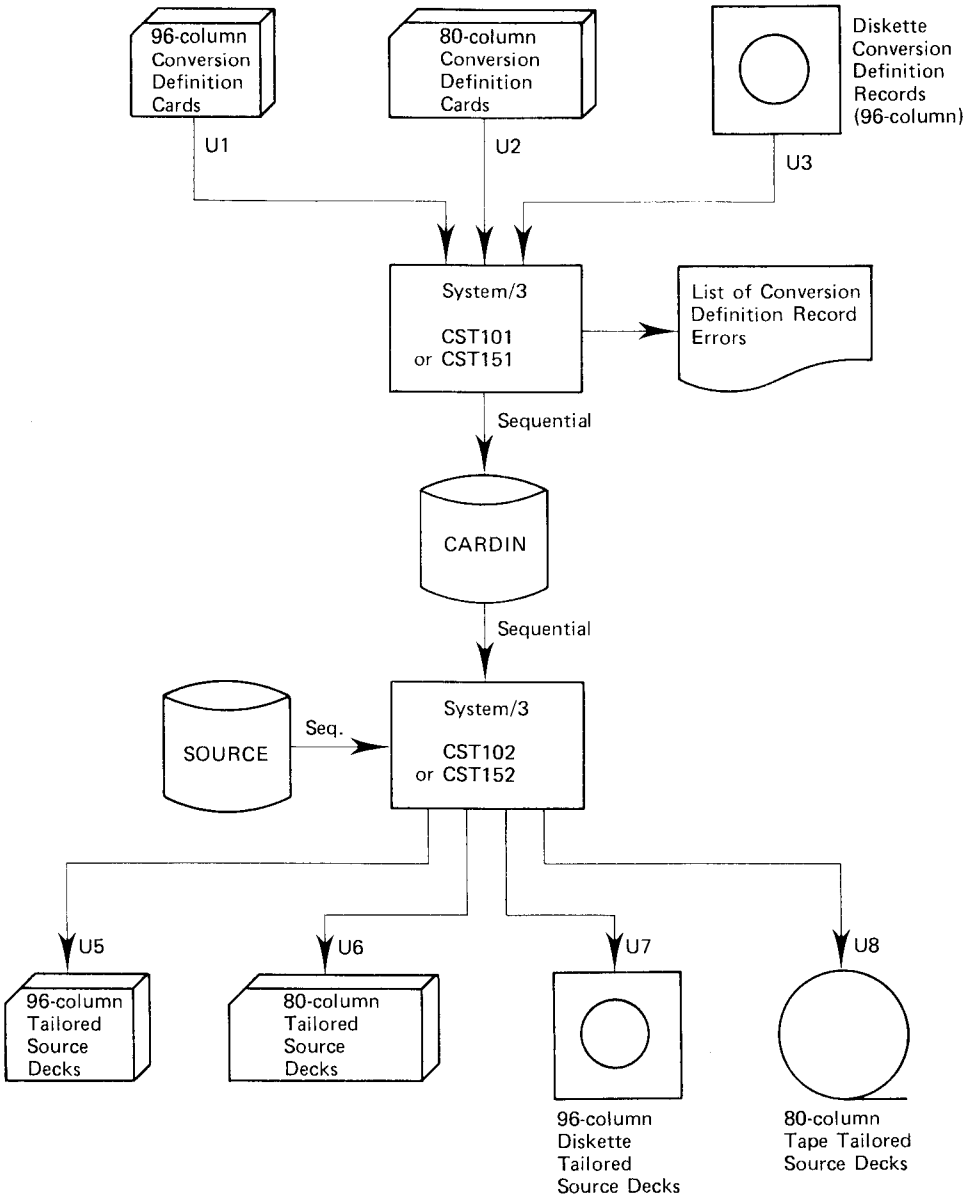
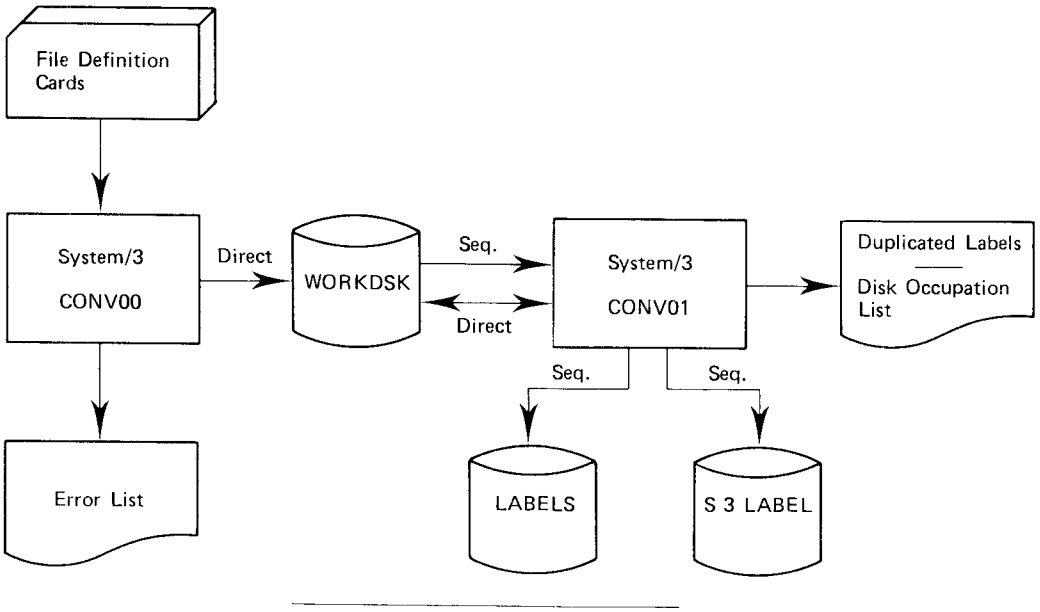


Figure 1 (1 of 5). Program Flow

FILE DEFINITION STEPS



CREATING THE PROGRAM FILE

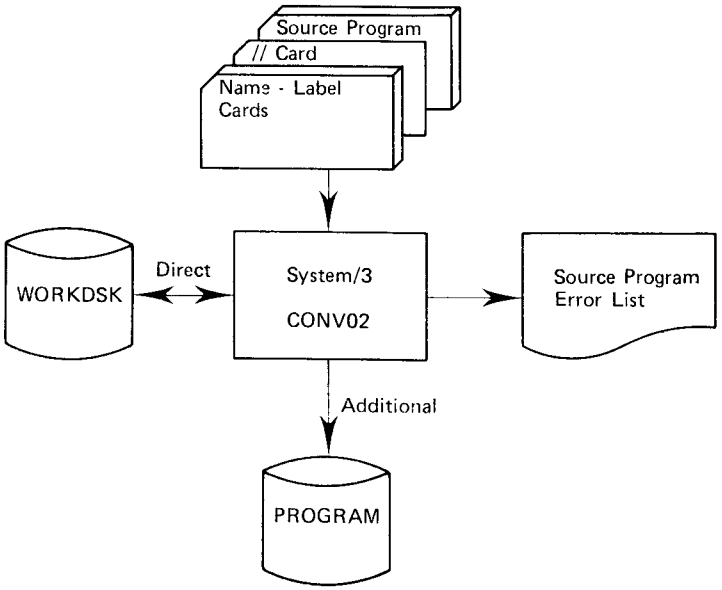
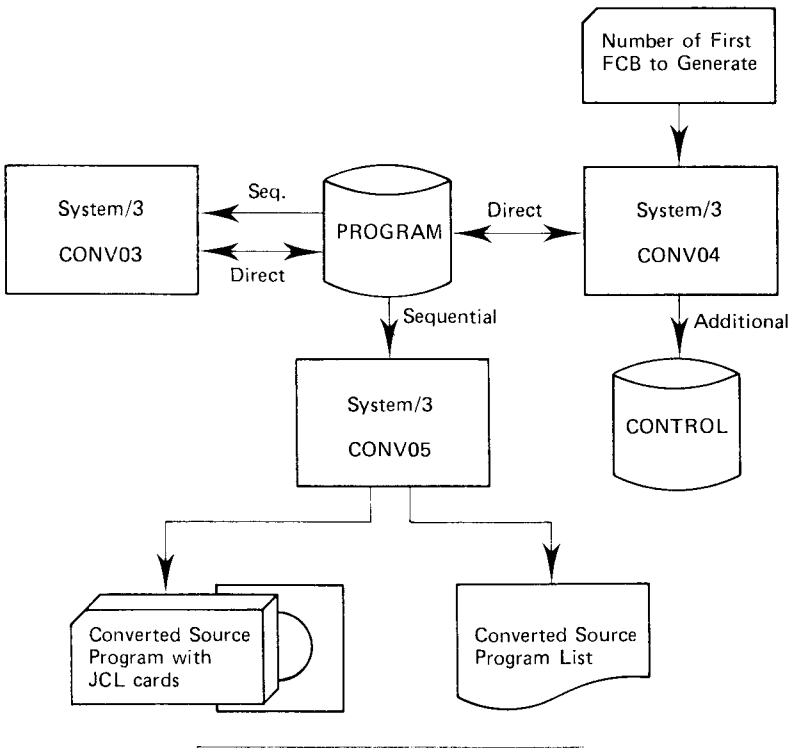


Figure 1 (2 of 5). Program Flow

SOURCE PROGRAM CONVERSION



GENERATING CARRIAGE CONTROL FCBs

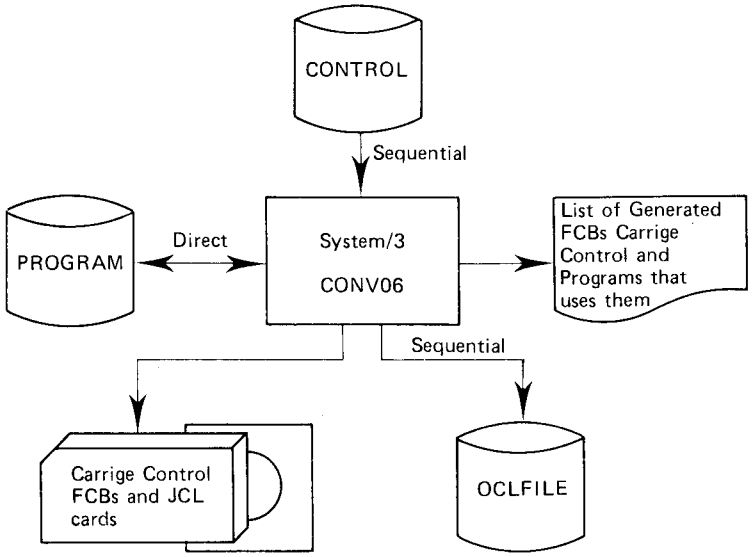


Figure 1 (3 of 5). Program Flow

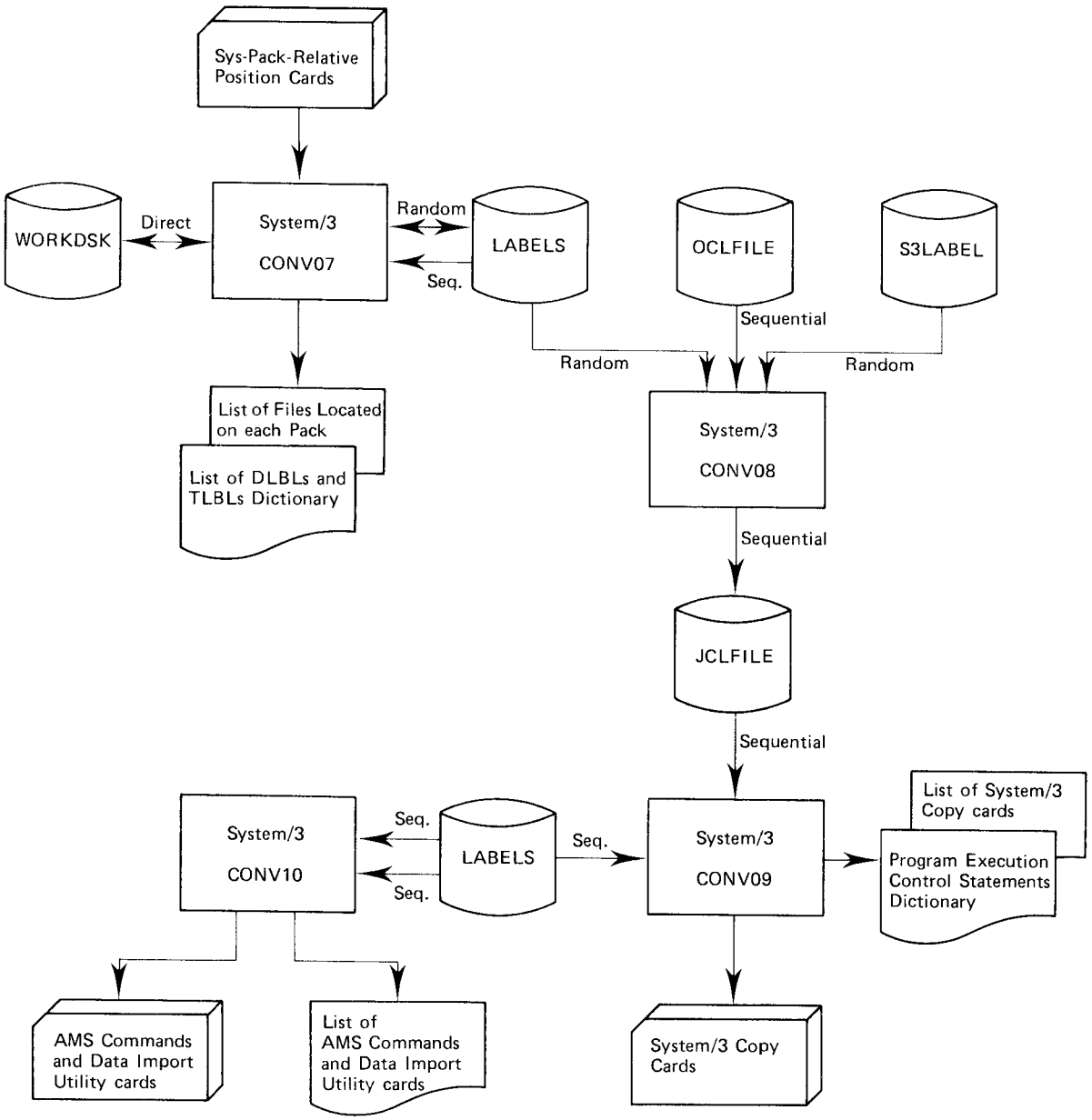


Figure 1 (4 of 5). Program Flow

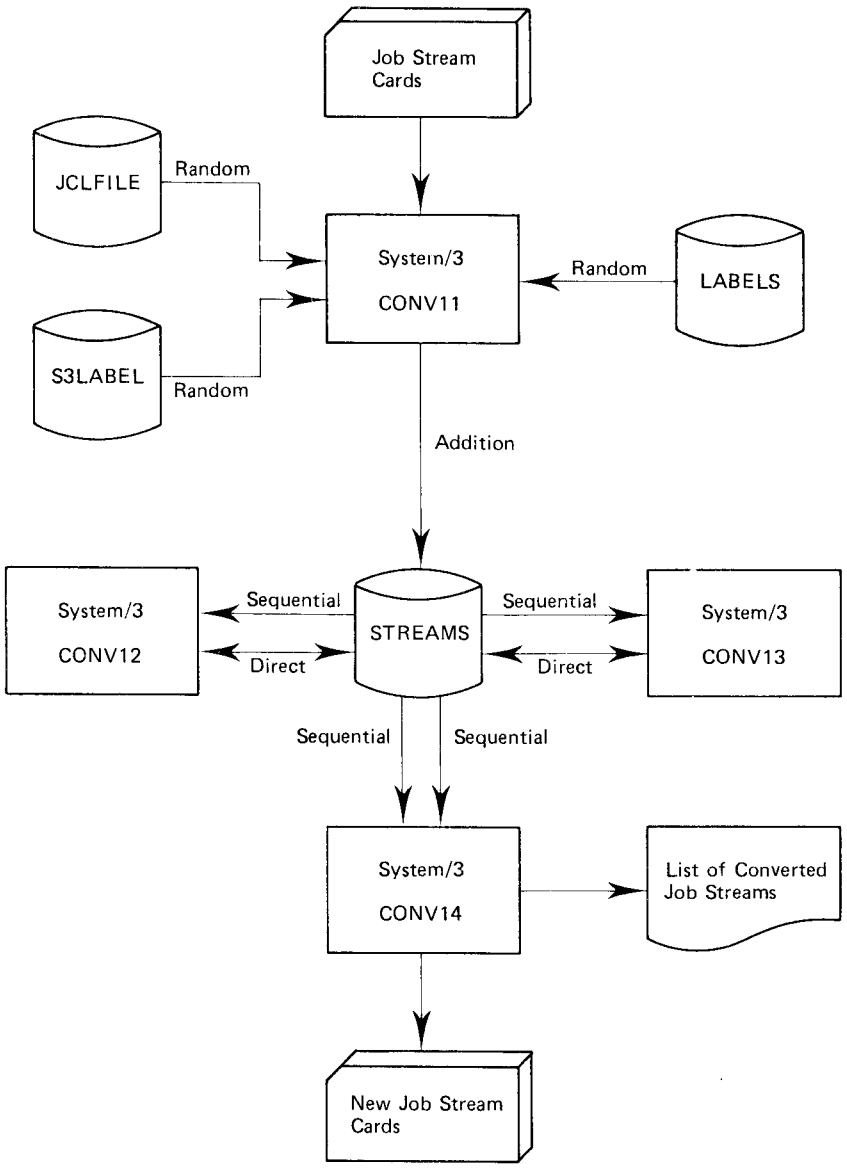


Figure 1 (5 of 5). Program Flow

INSTALLING THE PROGRAM

The magnetic tape containing the System/3 to DOS/VSE Conversion Aid International Field Program may be restored directly onto an IBM 5444 Disk Storage Drive, or into the 5444 simulation area of a 3340.

When the disk has been restored, it will contain:

- An object library in which are cataloged the CST101 and CST102 components (for a System/3 Model 10) and the CST151 and CST152 components (for a System/3 Model 15B). These components make up the customizer, according to the System/3 model on which they are to be executed. The SUBR93 subroutine is cataloged in the R Sublibrary.

- A source library, in which is cataloged the source CST151 and CST152 components to be compiled in case a System/3 Model 15-C or 15-D, is used in the customizer step.

- A file containing all the conversion components and the routine to be cataloged in the System/370 or 4300 processor.

CUSTOMIZING THE PROGRAM

The customizer components are executed only once and modify the definitions of the different files used for conversion. In addition, they retrieve the conversion components from the 5444 disk and place them, together with the routines to be cataloged in DOS/VSE Relocatable Library , onto the medium specified in the SWITCH statement. These components can then be compiled and cataloged in the System/3 used for the conversion process.

Output can be on 80- or 96-column cards, diskette, or magnetic tape.

CATALOGING MODULE IN THE DOS/VSE RELOCATABLE LIBRARY

The input job stream obtained as trial output from the customizer is a DOS/VSE SYSIN file that assembles and punches the WKVWXV module with the JCL to catalog it in the Relocatable Library. This module is written in DOS/VSE Assembler.

Once the SYSIN has been executed, the obtained deck needs be executed also.

SAMPLE PROBLEM

The complete execution of the sample problem distributed with the program is given in Appendix C which also contains a listing of the sample problem as follows:

- Input:
  - Statements containing specifications for customization
  - Input statements (OCI control statements, source programs, and data).

- Output:
  - Converted source programs
  - FCBs to be cataloged
  - System/3 OCI statements and DOS/VSE JCL statements to be used to convert disk files, using magnetic tape and Data Import Utility as an intermediate medium, and to define VSAM catalog and data spaces.
  - Generated DOS/VSE JCL statements.
- Listing obtained after having executed the sample problem.

After the source programs have been sustomized and cataloged, the actual conversion process can be started. Components CONV06 to CONV10 should be executed, if possible, only once during the conversion process in order to keep the number of printer carriage control tapes to a minimum and to avoid unwanted dualities in the allocation of SAM files.

If the components CONV06 to CONV10 are executed a second time:

- Make sure that SAM files located on each pack used in previous processes are exactly equal to the ones located in the second time. If this condition is not true, the automatic location assigned to SAM files in the second process can be different.

- Delete the existing CONTROI file before executing the CONV04 component to avoid adding data to the end of existing data.

- Enter the number of the first FCB not used in the previous conversion process in place of the number 000 or 001 for the CONV04 component when re-executed.

If the compcnents CCNV00 and CONV01 are executed a second time, re-enter CONV00 control statements containing specifications of all files to be used in the conversion, even though some of the information may have already been entered.

The CONV02 component is the only one that has to be executed for each source program to be converted. Components CONV03 to CONV06 are executed only once for all programs that have been accepted by the CONV02 component.

The converted programs can refer to all or part of the files entered using the CONV00 component. If they refer to only a part of the files, components CONV03 to CONV05 will have to be executed several times, once for each grcup cf programs that refer to the files. The data for the different groups is added to the CONTROL file. At the beginning of each group of source programs, and before executing the CONV02 component, the PROGRAM file must be deleted to avoid adding programs following the previously converted ones.

Components CONV11 to CONV14 may be used as often as needed to convert job streams, as long as these job streams refer to files and programs described in the JCLFILE, S3IABEI, and LABEIS files. The CONV11 to CONV14 components do not update these files.

The input job streams may not contain CALL statements to request cataloged procedures. All control statements for the program must be included in the job stream.

INPUT

To customize the source program before beginning the conversion process, the following data must be entered for the CST101 or CST151 component, depending on the System/3 model used and on the I/O medium indicated in the SWITCH statement:

- A CNV record indicating the configuration of the System/3 on which the conversion is to be performed

- A FUT record indicating the configuration of the System/370 or 4300
- As many EQU records as required to specify the equivalences between System/3 devices used by the programs and the devices in the System/370 or 4300 configuration.

As shown in Figure 1, file definition data (one record per file) must be provided for the CONV00 component. This and all other input must be on the input medium specified for the customizer. NAME-LABEL cards must be placed before the source program in the CONV02 component, followed by a // record. In all executions of the CONV04 component applying to one run of the CONV00 and CONV06 components, the first unused FCB must always have the same number. If the numbers are different, the number in the last execution of CONV04 is assumed.

As input to the CONV07 component, the relationship between the physical disk units and the logical SYSXXX units must be specified. At the same time, and in the same statement, the Relative Position in which the disk space of the pack is allocated must be specified.

The job streams to be converted are submitted as input to the CONV11 component.

FORMAT OF CONTROL STATEMENTS

CONV00 Control Statements for File Definition

POSITION OR COLUMN	IDENTIFICATION	DESCRIPTION
1	Identifier	F
2-30	Label	System/370 file identification. In VSAM, the file identification must be coded as follows: • One to 29 characters: A to Z, 0 to 9, -, #, @, or $ • A period must follow each group of eight or fewer characters • No embedded blanks are allowed • The first character of each group must be alphabetic • The last character of the file identification cannot be a period. For other files, the first character must be alphabetic. Valid characters are A to Z, 0 to 9, blank, -, ., #, @, and $.
31-34	Block length	(For tape files) Block length must be a multiple of the record length.
35-38	Record length	A number from 3 to 4096
39	File type	S sequential V SAM VSAM managed T tape K VSAM KSDS R VSAM RSDS E VSAM ESDS A VSAM ESDS with alternate index
40-41	Key length	(For I, K, or A files) A number less than or equal to 29, and less than or equal to the record length
42-45	Key location	For I, K, or A files

46-47	Percent of free space in control intervals	For K files
48-49	Percent of free space in control areas	For K files
50-52	Logical SYSXXX units	A number from 001 to 099
53-56	Validity period in days	If omitted, zero is assumed
57-62	Estimated number of records	Based on this number, the size of the file is calculated
63-64	Initials of the application	The CONV01 listing of file occupancy is printed, sorted according to these initials
65	Destination	Mandatory field. To generate REPRO statements to create a VSAM file or DITTO statements to create other files, an R must be entered. To generate Data Import Utility statements to create other files a U must be entered. Otherwise, an N must be entered.
66-73	System/3 label	System/3 file identification

CONV02 NAME-LABEL Statement

POSITION OR COLUMN	IDENTIFICATION	DESCRIPTION
1-8	File name	System/3 file identification
9-37	Label	DOS/VSE Label identification

CONV04 Statement Containing First FCB Number

POSITION OR COLUMN	IDENTIFICATION	DESCRIPTION
1-3		Number of first FCB

CONV07 SYS-PACK Relative Position Statement

POSITION OR COLUMN	IDENTIFICATION	DESCRIPTION
1-3	Logical unit	SYSnnn, corresponding to positions 50 to 52 of the CONV00 statement
4-9	Pack name	Volume identification of the pack in which files assigned to the SYSnnn will be allocated.
10-16	Relative Position	Number of relative track/block positions in which space allocated on the pack must start.

Customizer (CST101 or CST151) Input

```
// SWITCH 10000000     Control statements on 96-column cards
// SWITCH 01000000     Control statements on 80-column cards
// SWITCH 00100000     Control statements in 96-column diskette records
```

Customizer (CST102 or CST152) Output

```
// SWITCH 00001000      Program output on 96-column cards
// SWITCH 00000100      Program output on 80-column cards
// SWITCH 00000010      Program output on diskette
// SWITCH 00000001      Program output on magnetic tape
```

In the System/3 UPSI, only one of the bits U1, U2, and U3, and only one of the bits U5, U6, and U7 may be set. The U8 bit may be set at the same time as one of the U5, U6 or U7 bits.

CST101 or CST151 CNV Statement

POSITION OR COLUMN	IDENTIFICATION	DESCRIPTION
1-3	CNV	Identifier
4-10	MFCU1 READ01 READ42 MFCM1 DISKET	Conversion input device
10-17	DISK DISK45 DISK40	Conversion work files
18-24	MFCU2 TAPE MFCM2 READ42 DISKET	Device for output of CONV05 and CONV06 components
25-31	MFCU2 TAPE MFCM2 READ42 DISKET	Device for output of CONV09, CONV10, and CONV14 components
32-36		Number of records in the WORKDSK file

CST101 or CST151 FUT Statement

POSITION CR COLUMN	IDENTIFICATION	DESCRIPTION
1-3	FUT	Identifier
4-7	3330 3340 FBA	System/370 or 4300 disk units
8-9	18 to 99	Default forms length
10-11	14 to 96	Default overflow line
12	Y or N	Use of POWER/VS
13-18	3203-4 3211 3203 5203 1403 3203-5 PRT1	System/370 or 4300 printer

CST101 or CST151 EQU Statement

POSITION OR COLUMN	IDENTIFICATION	DESCRIPTION
1-3	EQU	Identifier
4-10	MFCU1	System/3
	MFCU2	
	PRINTER	
	PRINT84	
	CONSOLE	
	blank	
	DISK	
	DISK45	
	DISK40	
	TAPE	
	SPECIAL	
	CRT77	
	READ01	
	MFCM1	
	MFCM2	
	READ42	
	DISKET	

POSITION OR COLUMN	IDENTIFICATION	DESCRIPTION
11-17	MFCU1	System/370 or 4300
	MFCU2	
	PRINTER	
	CONSOLE	
	blank	
	DISK30	
	DISK40	
	TAPE	
	SPECIAL	
	READ01	
	READ05	
	READ20	
	READ25	
	READ40	
	READ42	
	MFCM1	
	MFCM2	
	DISKET	
	FBA	
18-21		Maximum record length (required for printer and diskette)
22-24		Logical SYSXXX DOS/VSE device (required for all devices except DASD)
25-27		Physical device address (required for all devices except DASD; used to create ASSGN statements)

Notes:

1. The 5444 simulation area of a 3340 disk does not support random access to indexed files. Since the International Field Program uses random access to indexed files, this area cannot be used as a conversion work area. Therefore, positions 10 to 17 of the CNV statement can be used only for an actual 5444 disk unit.

2. If a System/3 card reader punch is replaced by two different DOS/VSE devices, two EQU statements must be used, with the same code in positions 4 to 10 and different ones in positions 11 to 17.

3. The first EQU statement read by the customizer will be considered as the definition of the input unit and the second as that of the output unit. If there is only one EQU statement, the code in positions 11 to 17 will be considered as the definition of the input/output unit.

PROCESSING

In the input to CONV00, there must be one F statement for each labeled tape file or disk file used by the programs to be converted. If CONV00 is executed without error, CONV01 will be executed; if not, the control statements must be corrected and CONV00 rerun.

When CONV00 and CONV01 have been executed without error, CONV02 must be executed for each program to be converted. The input data for CONV02 includes a NAME-IABEI statement for each labeled tape or disk file used by the program, followed by a // statement and the source program to be converted. If a program is rejected (an explanatory message is displayed at the printer), it will not be added to the file of converted programs and must be reconverted.

When all the source programs have been accepted by CONV02, CONV03 to CONV05 can be executed one after the other without interruption. Input to CONV04 will include the number of the first FCB not used by former runs of CONV06.

CONV05 produces the converted source programs, together with the control statements required to compile them and catalog them in the DOS/VSE. In addition, it produces a listing of the converted source programs, with the number of the corresponding error messages. The text of the messages is given at the beginning of each program listing.

Messages numbered 25 and above are warning messages, and no action is required. However, a special processing routine for sequential files will be cataloged in the DOS/VSE Relocatable Library in response to message 38 or 55.

Messages 1 to 25 require action of some kind. In some cases, the system response must be verified, and in others manual intervention is required.

The converted source programs must be modified according to the appropriate messages before being compiled on the System/370.

When CONV05 has been executed, CCNV06 is run to list the printer carriage control tape images and to prepare for cataloging the FCBs.

From the SYS-PACK RELATIVE POSITION statements, CONV07 produces a listig of file allocation per disk pack. The disk space is automatically allocated as follows:

- Are located SAM files (in Label alphabetical order) without any space between a file and the next one, they are allocated at the specified Relative Position.

- The disk space behind the one used by SAM files (if any) is dedicated to VSAM Data Space.

CONV09 and CONV10 provide the statements required to prepare the conversion of master files using data from the LABELS file.

CONV11 to CONV14 convert the System/3 OCL statements; the number of statements to be converted depends on the size of the STREAMS file. If CONV11 terminates on an unexpected /*, /&, or /., the remaining OCL statements must be deleted from the input file; otherwise, the System/3 will execute them as a job.

OUTPUT

The output from the conversion aid program includes listings, messages, and miscellaneous output.

LISTINGS

The customizer produces no listings, unless errors are detected in the input statements.

CONV00 produces a listing of control statement errors for file definition, in case the same data must be processed again. If the input data contains no errors, no listing is produced.

CONV01 produces a listing of all the files defined in each logical SYSXXX file. The listing contains the name of the file, the number of records in the file, and the occupancy of the file on disk.

CONV02 produces printed output only when errors are detected. If errors are detected, the source program being converted is not placed on disk.

CONV05 produces a listing of the converted source programs, together with the messages indicating modifications to be performed.

CONV06 provides a listing of all carriage control tape images, as well as a listing of the programs that use each control tape.

CONV07 provides either a listing of errors or a listing of the files in each disk pack and a listing of the DLBLs, EXTENTs, and TLBLs of all files. It lists DLBLs and TLBLs, leaving file names blank, since the file name may be different in each program using the file.

CONV09 lists a control statement dictionary to be used when executing the components of the program. It is assumed that the files used will be those specified in the NAME-LABEL statements for CONV02. The dictionary is classified by program name and includes:

- // ASSGN statements
- // PAUSE * MOUNT CARRIAGE TAPE xxx (for a 1403 printer without POWER/VS)
- // EXEC SYSBUFLD (with no 1403 printer and without POWER/VS)
- * $$ LST statement with FCB parameter
- DLBLs, EXTENTs, and TLBLs from the files to be used
- EXEC statement for the program
- AMS BLDINDEX statement if a VSAM ESDS file with alternate index is created.

CONV09 and CONV10 produce listings of converted statements to convert files.

CONV14 provides a listing of the generated DOS/VSE JCL statements, together with short comments. Unconverted System/3 OCL statements are included, with warning messages indicating why that conversion was not performed.

CONV00 Messages

LABEL MISSING
INVALID LABEL CHARACTER OR SEGMENT IABEL
INVALID BLOCK LENGTH
INVALID RECORD LENGTH
KEY LENGTH MISSING
INVALID KEY LENGTH
INVALID KEY LOCATION FOR THIS FILE
LOGICAL SYS MISSING
FREE SPACE FOR A NON KSDS FILE
INVALID DESTINATION FILE CODE
SYSTEM/3 LABEL MISSING
INVALID FIRST CHARACTER LABEL OR LABEL SEGMENT
BLOCK LENGTH MISSING
RECORD LENGTH MISSING
INVALID FILE TYPE
INVALID KEY LENGTH FOR THIS FILE
KEY LOCATION MISSING
INVALID KEY LOCATION
INVALID LOGICAL SYS
NUMBER OF RECORDS MISSING
INVALID RECORD LENGTH FOR DESTINATION R
SYSTEM/3 LABEL INVALID

CONV01 Messages

DUPLICATED IABELS
SYSTEM/3 DUPLICATED LABELS
PLEASE, CORRECT DUPLICATED LABELS AND REPEAT THE WORK

CONV02 Messages

MORE THAN 15 NAME-LABEL RECORDS
FILE NOT PREDEFINED
PROGRAM NAME INVALID
H CARD MISSING
INVALID FILE DEVICE
FILE IS NOT IN NAME-LABEL RECORDS
MORE THAN 15 FILES
FILE NAME REPEATED
MORE THAN ONE PRINTER FILE
FILE ATRIBUTES DO NOT MATCH
TABLE FILE CANNOT BE A KEYED FILE
MORE THAN ONE ADDROUT OR BY ADDRCUT FILE
ADDROUT WITHOUT FILE BY ADDRESS OR OPPOSITE
THERE IS NO VALID F CARD
ANY NAME-LABEL RECORD NOT USED
ENTRY IN TABLE GREATER THAN RECORD SIZE
TR RECORD MUST BE IN A CARD FILE
FIELD END POSITION GREATER THAN RECORD
SOURCE PROGRAM CARDS OUT OF SEQUENCE
THERE ARE NO C CARDS
ERRORS DETECTED IN SOURCE PROGRAM
-CORRECT IT, AND REPEAT THE STEP

CONV05 Messages

1 COMBINED FILES ARE USED - MODIFY PROGRAM LOGIC TO ELIMINATE IT.
2 CONSOLE FILES MUST BE OUTPUT OR DISPLAY - MODIFY PROGRAM LOGIC.
3 PRIMARY, RANDOM, DEMAND, DISPIAY, TABLE OR LIMITS FILES MUST NOT BE CONDITIONED TO U INDICATOR - ASSUME NOT CONDITIONED
4 SPECIAL FILE MUST NOT BE BLOCKED NOR TWO I/O AREAS.
5 THE FILE HAS ADDITIONS BUT NO ADDS ARE SPECIFIED IN O CARDS.
6 THE OBJECT TIME TABLE MUST BE MODIFIED TO FIT DATA IN NEW RECORD SIZE.
7 M-FIELDS ARE USED - MAY BE NECESSARY TO CHECK THE *ERROR FIELD.
8 SPECIAL SEQUENCE ARE SPECIFIED - MAY BE NECESSARY TO CHECK THE *ERROR FIELD.
9 NON IDENTIFIED RECORD CAN CANCEL THE PROGRAM - MAY BE NECESSARY TO CHECK THE *ERROR FIELD.
10 STACKER SELECTION SPECIFIED IN I CARD HAS BEEN SUPPRESSED - VALIDATE.
11 INVALID OPERATION - SUPPRESS IT.
12 REWRITE ASSEMBLER MODULE IN DOS/VSE ASSEMBLER.
13 MORE THAN 11 SKIP POSITIONS ARE SPECIFIED - MODIFY THE PROGRAM.
14 CARD PRINTING HAS BEEN CONVERTED TO COMMENT CARD - VALIDATE.
15 STACKER SELECTION SPECIFIED IN O CARD HAS BEEN SUPPRESSED - VALIDATE.
16 RESERVED WORD IS USED - MODIFY IT.
17 A BLDINDEX AMS COMMAND MUST BE EXECUTED AFTER THE EXECUTION OF THIS PROGRAM.
18 CONVERT THE SOURCE MODULE AND CATALOG IT IN THE R SUBLIBRARY WITH THE SAME NAME.
19 H INDICATORS ARE USED - THE PROGRAM CAN BE CANCELLED.
20 THE COMPILATION TIME TABLE MUST BE MODIFIED TO FIT DATA IN 80 COLUMNS
30 INQUIRY SUPPORT HAS BEEN SUPPRESSED IN H CARD.
31 1P FORM POSITIONING HAS BEEN SUPPRESSED IN H CARD.
32 DEBUG USE HAS BEEN SUPPRESSED IN H CARD.
35 FILE NAME HAS BEEN SHORTENED TO SEVEN CHARACTERS.
36 FILE NAME HAS BEEN MODIFIED.
37 NEW TABLE FILE.
38 INCLUDED MODULE TO ACCESS BY RELATIVE RECORD NUMBER AN ESDS FILE.
40 TABLE READ BY NEW TABLE FILE.
41 MODIFIED NUMBER OF ENTRIES PER RECORD IN OBJECT TIME TABLE.
42 MODIFIED NUMBER OF ENTRIES PER RECORD IN COMPILATION TIME TABLE.
43 THE COMPILATION TIME TABLE HAS BEEN MODIFIED TO REDUCE RECORD SIZE
45 L CARD HAS BEEN CONVERTED TO COMMENT CARD.
46 L COMMENT CARD HAS BEEN ADDED.
50 DECIMAL POSITION HAS BEEN ADDED TO AN ARRAY OR ARRAY ELEMENT IN I CARD.
55 INSTRUCTIONS ADDED TO BRANCH TO RANDOM MODULE.
56 CHAIN INSTRUCTION CHANGED TO CREATE AND RRDS RANDOMLY.
57 THE SQRT HAS BEEN HALF ADJUSTED.
60 OVERFLOW INDICATOR MUST NOT BE IN AND LINE - CHANGED TO PRIMARY LINE.
61 CHANGED SKIP TO A LINE NUMBER BY SKIP TO A CHANNEL.
62 ONE SPACE AFTER ASSUMED.
63 ADD HAS BEEN SUPPRESSED IN O LINE.
64 L0 INDICATOR HAS BEEN SUPPRESSED IN O LINE.
65 SPACES FOR CONSOLE LINE HAVE BEEN SUPPRESSED.

CONV07 Messages

ERRORS IN SYS - PACK RELATIVE POSITION CARDS
SYS USED BUT NOT DEFINED
SYS INVALID
SYS USED MORE THAN ONCE
RELATIVE POSITION MISSING
NON-EQUAL RELATIVE POSITIONS TO SAME PACK
SYS DEFINED BUT NOT USED
CORRECT CARDS AND REPEAT THE WORK

CONV14 Messages

FUNCTION NOT SUPPORTED
THIS STATEMENT HAS NO EQUIVALENCE IN DOS/VSE
NO INFORMATION ABOUT THIS PROGRAM
UNEXPECTED END OF STEP
ERROR IN O.C.I.
FILE WITH NAME OR UNIT MISSING
SORT OR COPY WITH ANY FILE-NAME MISSING
SORT OR COPY WITH UNLABELED TAPE FILE
FILE NAME REFERENCES ONE FILE NOT PREDEFINED
SORT HEADER CARD MISSING
TAPE FILE WIHOUT FILE NAME
TAPE SORTIN WITH MORE THAN ONE FILE NAME
SORTIN WITH DIFFERENT TYPES
OUTPUT RECORD LENGTH AND SPECIFIED RECORD LENGTH DO NOT MATCH
ALTSEQ SPECIFIED AND ALTSEQ RECORDS MISSING
MORE THAN ONE RECORD TYPE OR INCLUDE/OMIT SET, OR SEQUENCE ERROR
INCLUDE AND OMIT BOTH PRESENT
FORCED CONTROL FIELDS USED (F IN COLUMN 7)
DATA CHARACTER FORCED USED (V IN COLUMN 8)
ZONE PORTION USED (Z IN COLUMN 8)
DIGIT PORTION USED (D IN COLUMN 8)
SUM FIELDS WITH MODIFIED LENGTH USED
SORTRS WITHOUT SUM FIELDS OR VICEVERSA
SORT WITHOUT F SENTENCES
SORT ADDROUT NOT ALLOWED WITH SAM FILE
ERRORS IN COPY SPECIFICATIONS
OMIT OR DELET USED
SELECT PKI USED
COPYFILE STATEMENT BYPASSED
OUTPUT OTHER THAN DISK, FILE, PRINT OR BOTH
SELECT KEY WITH UNKEYED FILE
* * * EXECUTION * * *
* NOT CONVERTED *
  * * * * INSERT ANY DATA HERE * * * *
* * TEXT SHORTENED IN NEW JOB STREAM * *
* STATEMENT BYPASSED IN NEW JOB STREAM *

CST101 and CST152 Messages

CNV OR FUT CARD DUPLICATED.
---------DEVICE INVALID.
INVALID NUMBER OF RECORDS.
FORM LENGTH OR OVERFLOW LINE INVALID.
INVALID POWER SPECIFICATION.
S/3 DEVICE INVALID.
S/3 DEVICE DUPLICATED.
DOS/VSE DEVICE INVALID.
INVALID DEVICE CORRESPONDENCE.
INVALID MAXIMUM RECORD LENGTH.
INVALID LOGICAL SYS.
INVALID PHYSICAL ADDRESS.
CNV CARD MISSING OR FUT CARD MISSING.
READER DEVICE MISSING.
PRINTER DEVICE MISSING.
DISK DEVICE MISSING.
ONE U5-U8 INDICATOR ON NEEDED.
CARD OTHER THAN CONV. FUT OR EOJ
ONE U1-U3 INDICATOR MUST BE ON
ONLY ONE U5-U7 ON INDICATOR ALLOWED.
ANY ERROR DETECTED - REPEAT THE STEP.

MISCELLANEOUS OUTPUT

Customized Programs

CONV00 to CONV14 are placed on the output medium specified in the SWITCH statement. Each component is preceded by the statements:

```
// CALL RPG,R1
// RUN
```

The sample problem to be executed is placed on the same medium, following the last component.

The sample problem will be followed by the routines to be cataloged in the System/370 or 4300. If two bits are set in the SWITCH statement (one of these bits must be bit 8), the program components will be placed on cards or diskette and on magnetic tape.

Although programs cannot be cataloged in a System/3 from magnetic tape, the customizer supports tape output in order to speed up customization. The tape may be converted offline to a different medium.

Converted Programs

CONV05 places the converted source programs, together with the control statements needed to compile and catalog them, on the output medium specified in the customizer. An image of this output is produced as a listing.

Before the converted programs are compiled, system messages must be consulted and any required modifications or manual conversion performed.

Form Control Buffers

CONV06 produces job streams to be cataloged on the DOS/VSE C. I. Library. The phases correspond to the FCBs generated by CONV06, except when the carriage control tape is for a 1403 printer and POWER/VS is not used.

Aids for File Conversion

For files to be copied from System/3 to System/370 or 4300 via tape, CONV09 generates:

- $TINIT statements to initialize the tape reels.
- COPY statements to copy from disk to labeled tapes, using the System/3 $COPY program.

For VSAM files, CONV10 always generates:

- AMS statements to define the master catalog. The name of the disk pack must be added to the EXTENT statement and to the VOLUME parameter of the AMS statement. The starting track/block of the file must be added to the EXTENT statement.
- AMS statements to define data spaces.
- AMS statements to define clusters and the alternate index and its corresponding path in ESDS files with alternate index.
- AMS statements to convert from tape to disk with REPRO and BLDINDEX statements for ESDS files with alternate index (if file conversion was requested in CONV00 with an R in column 65).

For SAM and SAM VSAM managed files, CONV10 generates:

- AMS statements to define managed non VSAM files.
- AMS statements to convert from tape to disk with REPRO (if file conversion was requested in CONV00 with an R in column 65).
- Data Import Utility statements to convert from System/3 Disk 3340 to 4300 Disk. (If file conversion was requested in CONV00 with a U in column 65).

Normally, only permanent data files (master files, tables, and so on) are converted; intermediate or work files are not.

Job Streams

The output from CONV14 is a DOS/VSE job stream containing all JCL statements needed to execute source programs, sorts and REPRO (AMS).

To execute the SORT program, DOS/VSE must contain the standard definition of the SORTWK1-SYS001 work area. In addition, if not enough real storage is available for the AMS BLDINDEX, DOS/VSE must contain the standard definition of the IDCUT1 and IDCUT2 areas.

WORK FILES

The WORKDSK file is created by CONV00 and contains information on all user files.

It is used by CONV01 and CONV02 and then by CONV07. The LABELS file is created by CONV01 and is used by CONV07, CONV08, CONV09, CONV10, and CONV11.

The PROGRAM file contains a record for each statement in the source programs being converted. The records are preceded by a record containing pointers to the address of the first statement of each type. The PROGRAM file is created during the first execution of CONV02. During the following runs, records are added to the file. After having been used by CONV03 to CONV05, the file is no longer needed and is used by CONV06 as a work file for CONTROL file records.

The CONTROL file, created and completed by CONV04 and used by CONV06, contains a record for each source program file, plus a record for each printer file. The CCLFILE file, created by CONV06 and used on request by CONV08, contains a record for each source program file.

The S3LABEL file is a table that indicates the relationship between System/3 labels and DOS/VSE labels. It is created by CONV01 and is read at random by CONV08 and CONV11.

The JCLFILE file has three types of records. One contains information needed to create ASSGN statements, another the information needed to create DLBL and EXTENT statements, and the third contains card images. The file is created by CONV08 and used by CONV09 and CONV11.

The STREAMS file is created by CONV11 from the input job streams to be converted and is updated by CONV12 and CONV13. It serves as input to CONV14.

Detailed information on the contents of these files is given in Appendix B.

SIZE OF WORK FILES

The size of the different work files is shown in Figure 3.

EXECUTING THE CONVERTED PROGRAMS ON THE SYSTEM/370

When the converted programs have been modified, if it is necessary (see CONV06 messages 1 to 29) they must be compiled using the appropriate RPG II compiler. Before being link edited (if CONV05 messages 38 or 55 have been printed), the WKV routine must be cataloged.

FILE	LENGTH (IN BYTES)	NUMBER OF RECORDS
WORKSDK	128	One for each source program file, plus one for each statement of the longest source program to be converted, plus 20
PROGRAM	128	One for each source statement of the program to be processed in one run by CONV03 to CONV05, plus 20 for each program
LABELS	128	One for each defined file, plus one for each volume containing VSAM files
CONTROL	128	16 for each converted program
OCLFILE	64	Same number as the CONTROL file
S3LABEL	64	One for each defined file
JCLFILE	128	Difficult to estimate; however, 45 times the number of programs to be converted will be enough in most cases
STREAMS	128	Three times the number of statements in the job stream will be enough in most cases

Figure 2. Size of Work Files

The master files must be created in the System/370 or 4300 with or without the use of the conversion statements produced by the program.

Finally, the execution job stream produced by CONV14 (completed, if necessary) is executed on the System/370 or 4300.

THE CONV00 COMPONENT

The CONV00 component creates the WORKDSK direct access file and reads file definition statements. The input consists of file definition records, and output is the WORKDSK direct access file and an error listing.

File definition records of labeled disk or tape files are stored on the direct access file. File definition records are read and checked for errors; if no errors are detected, the records are placed on the WORKDSK file together with information on disk space, size of control intervals for VSAM files, and so on.

When CONV00 and CONV01 have been executed without error, they need not be executed again. Errors checked in file definition statements include:

- Missing labels
- First character of label or label segment not valid
- Invalid character in label or label segment
- Missing block (tape files)
- Invalid block (non-tape file or block length not multiple of record length)
- Missing record length
- Invalid record length (not between 3 and 4096)
- Invalid file type (not S, V, T, K, R, E or A)
- Invalid key length (greater than 29 or greater than record length)
- Missing key location (K or A file)
- Invalid key location (E, R, S, V or T file)
- Invalid key location (does not correspond to record length and key length)
- Invalid logical SYSXXX (100 or greater)
- Available space for non-KSDS file
- Number of records missing
- Creation code invalid (neither N nor R or U)
- Missing System/3 label
- Invalid System/3 label
- Missing key length
- Missing SYSXXX.

THE CONV01 COMPONENT

The CONV01 component checks duplicate labels and creates the LABELS file. The input consists of the WORKDSK and WORKDS1 direct access files, and the output is the WORKDS1 direct access file, the LABELS indexed sequential file, the S3IABEL indexed sequential file, and a listing of files, file sizes, duplicate labels, duplicate System/3 labels, and logical SYSXXX devices used.

If two names are found in a file, the first is the name given to the file definition statement, and the second is the disk label (if different from the first name).

CONV01 uses the records of the WORKDSK file according to two file definitions, one sequential and one random. It begins by sorting the records by label, in order to find duplicate labels and to create the indexed IABELS file.

Records are then sorted according to:

- Initials of the application
- Logical SYSXXX device
- File type
- Label.

This is done in order to set up a file list and to establish disk occupancy, both in the desired order.

The component lists the files within each application that are used by logical SYSXXX devices. It then sorts the records again by System/3 label to check for duplicates and to create the indexed S3LABEL file.

Finally, CONV01 produces a listing of the logical SYSXXX devices used.

If duplicate labels are detected, they should be corrected and both CONV00 and CONV01 executed again.

THE CONV02 COMPONENT

The CONV02 component reads and checks System/3 source programs; if no errors are detected, the source programs are added to the PROGRAM file. The input consists of NAME-LABEL statements, a // statement, and statements of the source program plus the WORKDSK direct access file. The output is the WORKDSK direct access file, the PROGRAM file containing all the source programs to be converted, and an error listing.

CONV02 must be executed without error, once for each source program to be converted. It reads the source program, preceded by NAME-LABEL records and the // record, and places it on the WORKDSK file, checking for possible errors. If no errors are detected, the source program is added to the PROGRAM file.

For each source program, the following records are created in the PROGRAM file:

- Type 1 record: pointers to the program
- Type 2 record: names of the files used in the program
- Type 3 records: one for each NAME-LABEL record read
- Type 4 records: one for each source statement read

- Type 5 records: one for each source statement added to the PROGRAM file (new table files, I statements, new I statements, new C statements, and table expansion during compilation)
- Type 6 record: /* statement
- Type 7 record: for U statements or Auto Report /COPY statements.

Source program errors that make conversion impossible include:

- More than 15 NAME-LABEL records
- NAME-LABEL record referring to an undefined file
- Disk file with no NAME-LABEL record
- Attributes of disk or tape file do not match defined file
- H statement missing
- Program name invalid
- File name repeated
- More than 15 files
- More than one printer file
- Invalid device for a file
- More than one addrout or by addrout file
- Addrout without file by addrout or viceversa
- There is no valid F statement
- NAME-LABEL record refers to a file not used in the program
- Final field position greater than record length
- Source statements out of sequence
- O statements missing
- Table file cannot be a keyed file
- TR records must be in a card device
- Length of entry in compilation table greater than 80.

If errors are detected, they should be corrected and the job repeated.

THE CONV03 COMPONENT

The CONV03 component partially modifies programs in the PROGRAM file. Input includes the sequential PROGRAM file and the PROGR file, the PROGRAM file with random access by relative record number. Output is the updated PROGR file.

CONV03 makes the following modifications, as necessary:

- Enlarges the first ten compilation tables if data occupies more than 80 bytes per record
- Creates new table files when more than one execution table is assigned to a file
- Modifies the E statement from execution tables or arrays in case changes are made to the number of entries per record, indicating that the table has to be modified
- Places NS in positions 15 and 16 of the look-ahead record
- Deletes stacker selection specified in I statements
- Checks for more than 11 skip positions
- Shortens or otherwise modifies the names of the files and places the new names in the F, E, I, C, and O statements
- If record length is changed, updates the E statements with the number entries per new record
- Converts I statement to comments or, if there is a printer file creates a new I comment statement
- In O statements, changes the skip to channel number
- Assumes one space after printing in printer lines without space or skip
- Erases ADD character in additions to ESDS files
- Changes the overflow indicator in C statements from AND line to primary line
- Detects and flags additions files that are defined when no record additions are specified for the file
- Issues a warning message when sequence control of records is performed
- Issues a warning message when identification checking is performed
- Converts printer commands to write on cards to comment statements and issues a message
- Issues warning messages about the use of reserved words in E, I and O statements
- Issues warning messages about the use of H indicators in I and O statements
- Removes IO indicators from C statements and issues a warning message about their use

- Deletes spaces from the output lines to console files
- Checks indicators used, and locates one unassigned indicator
- Relocates pointers to indicate the beginning of files rather than the beginning of programs.

THE CONV04 COMPONENT

The CONV04 component partially modifies source programs in the PROGRAM file. Input includes the CARD file with a record indicating the number of the XXXX FCB to generate and the sequential PROGRAM file with random access by relative record number. Output is the updated PROGRAM file and the sequential CONTROI file.

CONV04 makes the following modifications, as necessary:

- Removes options from the H statement that are not used in DOS/VS RPG II
- Changes file definition F statements according to the characteristics of the files
- Flags the console file if it is not an output or display file
- Flags combined files and issues a message to warn the user that the program should be modified to avoid them
- Primary, random, demand, display, table/array, and limit files cannot be conditioned by external indicators. References to external indicators are flagged, and the corresponding files are not affected.
- Flags the use of a dcuble I/O area or a block in special files
- Requests rewriting the module used by special files
- Issues a warning message when an IMS BLINDEX command must be executed following program execution, if an ESDS file is created with an alternate index
- Changes record length in ADDROUT files
- Flags the inclusion of an access module by relative record number for VSAM ESDS files
- Updates the added C statements to branch to the routine for random processing for VSAM ESDS files
- Updates C statements to create random access RRDS files
- Places decimal positions in the arrays or array elements defined in I statements
- Places an H in position 53 (half adjust) for SQRT operations
- Detects and flags invalid operation codes
- Issues a warning message when an EXIT to a routine in System/3 Assembler language must be rewritten in DOS/VSE Assembler language.

- Issues warning messages about the use of H indicator in C statements
- Issues warning messages about the use of reserved words in C statements
- Resumes warning messages used in each program and updates with this information the record type 2.

The component creates records and adds them to the CONTROL file, together with information to be used later. This information includes a type A record for each file defined and a type B record for each printer file with a forms definition.

The CARD file contains the number of the first FCB to be generated. Its purpose is to prevent creating several FCBs with the same name, when programs are converted in groups.

THE CONV05 COMPONENT

The CONV05 component places the converted source programs on the specified output medium and lists them on the printer. Input is the sequential PROGRAM file, and output is the SOURCES file of source programs on 80- or 96-column cards, diskette, or magnetic tape and listings of the source programs and detected errors.

CONV05 reads the PROGRAM file, places the source program statements in the SOURCES file with the JCL statements for compilation and link editing, and lists them on the printer. If the use of POWER/VS is specified, the component also creates JECL statements.

Warning and action messages for each program are also listed.

THE CONV06 COMPONENT

The CONV06 component creates FCBs, lists them, and indicates the programs in which they are used. Input is the sequential CONTROL file and the PROGRAM work file, and output is the PROGRAM work file, the SCURCE file of FCBs on 80- or 96-column cards, diskette, or magnetic tape, the indexed sequential CCIFILE file of execution control cards, and a listing of the created FCBs and the programs in which they are used.

The CONV06 component performs the following functions:

- Reads the CONTROL file and places its contents on the PROGRAM file, using the PROGRAM file as a work file throughout execution
- Sorts data in the PROGRAM file by record type, form length, overflow line number, and line channel numbers 1 to 11
- Reduces to a minimum the number of FCBs to be generated
- Creates form images in source program form and produces the JCL and JECI statements required to assemble and link edit them
- Lists the created carriage control tape images and indicates the program in which they are used
- Sorts data by program name, order number, and relative record number

- Creates the ordered indexed sequential OCLFILE file
- Numbers the created FCBs, starting with the number specified in the input to the CONV04 component.

THE CONV07 COMPONENT

The CONV07 component distributes defined files in disk packs and assigns the first relative track/block to SAM files. Input is the assignement statements relating the logical SYSXXX devices used in CONV00 to the disk packs and to the relative position in which the disk space must be located on each pack and the WORKDSK file of file definitions sorted by System/3 label. Output is the WORKDSK work file, the indexed LABELS file with the disk location of files and data spaces and two listings; the first one is a list of files distributed among the disk packs, the second one is a listing of DLBLs and TLBLs.

CONV07 reads SYS-PACK-REL. POSITION statements and if an error is detected in these statements, the component should be executed again.

THE CONV08 COMPONENT

The CONV08 component creates JCLFILE file. Input is the sequential OCLFILE file of System/3 CCL statements, the indexed LABELS file of file definitions and the indexed S3LABEL file of relationships between System/3 and DOS/VSE labels. Output is the indexed JCLFILE file of DOS/VSE execution control statements.

CONV08 creates the JCLFILE file containing JCL statements required to execute each converted program. The file is indexed, using as key the name of the program plus three digits (from 000 to nnn).

THE CONV09 COMPONENT

The CONV09 component produces a listing of JCL statements required to execute the converted programs and System/3 COPY statements with the corresponding listing of these statements.

Input is the indexed JCLFILE file of execution JCL statements, and the indexed LABELS file of file definitions, and output is a listing of JCL statements, the COPY statements and a listing of the generated statements.

The JCL statements are classed by program name and include:

- // ASSGN statements
- FCB printer statements, depending on the printer to be used and on whether POWER/VS is to be used:
  - // PAUSE * MOUNT CARRIAGE TAPE xxx (for 1403 with POWER/VS)
  - // EXEC SYSBUFLD (no 1403, no POWER/VS)
  - * $$ LST (no POWER/VS).

The COPY statement, generated to those files defined with an R in column 65 in CONV00, include:

- The $TINIT statements needed to initialize the tapes

- The System/3 $COPY tc copy disk sequential or indexed files to labeled tape files with a block length multiple of record length, but not greater than 4096.

THE CONV10 COMPONENT

The CONV10 component generates AMS commands and Data Import Utility statements and their corresponding listings.

CONV10 processes the indexed IABELS file of files definitions twice sequentially and generates:

- Master Catalog definition with the following data missing:
  - First relative track/block of the EXTENT
  - Pack name of the EXTENT
  - Pack name in AMS (data for the VOLUME keyword)
- Data Space definition
- Definition of clusters
- Definition of alternate index and its corresponding path for ESDS files with type A
- If the file definitiон statement used by the CONV00 component contained an R in position 65, it generates REPRO statements to copy the file from tape to disk. If the file is an ESDS type A, it creates the corresponding BUILD INDEX statements
- If the file definition statement used by the CONV00 component contained a U in position 65, it generates Data Import Utility statements.

THE CONV11 COMPONENT

The CONV11 component reads System/3 job streams and places them in the STREAMS file. Input includes the CARDS file of System/3 OCL statements, and the random access JCLFILE, S3IABEL, and LABELS files. Output is the STREAMS file of partially converted OCL statements.

CONV11 records each System/3 OCL statement in its original format and creates the corresponding DOS/VSE job control statement. Sort and copy statements are recorded in their original System/3 formats, to be converted in later steps.

/*, /&, and /. statements must be changed to */, &/, and ./ before executing CONV11 in order to avoid an unwanted end-of-step. If an EOJ statement is not changed, CONV11 can be executed again using the data cards not yet processed.

CONV11 builds DLBL and TLBL statements from information in the following files:

- JCLFILE, statements needed to execute each program
- S3LABEL, linkage to the LABELS file
- LABELS, information concerning files for programs executed using other files.

THE CONV12 COMPONENT

The CONV12 component converts System/3 sort programs to DOS/VSE sorts. This is done by processing the STREAMS file created by CONV14.

Input is the random access STREAMS file, and output is the same file updated with the necessary job streams.

A System/3 sort program cannot be converted if one of the following conditions exists:

- Errors in OCI statements
- References to an undefined file
- Name of input or output file omitted
- Specified and actual output record length not the same
- ALTSEQ specified and corresponding records missing
- More than one record type or INCLUDE/OMIT set
- INCLUDE and OMIT both present
- Forced control fields (F in position 7)
- Forced data character (V in position 8)
- Zone portion used (Z in position 8)
- Digit portion used (D in position 8)
- References to a keyed outupt file
- Summary Sort without SUM fields or SUM fields in non-summary sort
- Sort without F statements.

In order to execute the ccnverted sort programs, standard label information must be cataloged for work files and permanent logical device assignments should be made:

```
// DLBL SORTWK1,0,SD
// EXTENT SYS001,...
ASSIGN SYS001,...
```

If a numeric field is defined as D, it must be changed to C or U to be converted; sorts using the digit portion are not converted.

THE CONV13 COMPONENT

The CONV13 component converts System/3 copy programs to DOS/VSE AMS REPRO command. Input is the random access STREAMS file, and output is the updated STREAMS file.

CONV13 processes the STREAMS file created by CONV11 and CONV12.

A System/3 copy program cannot be converted if one of the following conditions exists:

- Errors in OCL statements
- References to an undefined file
- Processing of unlabeled tape file
- Name of COPYIN file omitted
- Error in copy specifications
- OMIT or DELET option used
- Select PKI used
- COPYFIIE statement omitted
- Output file other than disk, printer, or both
- COPYO file name omitted for output file.

AMS REPRO is used to copy files.

THE CONV14 COMPONENT

The CONV14 component creates the DOS/VSE job stream and lists it. Input is the sequential STREAMS file, and output is the DOS/VSE job stream statements and a listing of all statements, converted and unconverted. Unconverted statements are accompanied by comments to explain why they were not converted. Each step read by CONV14 is listed.

To execute CONV14, the STREAMS file must have been created and updated by CONV11, CONV12, and CONV13.

Positions 1 and 2 of each record of the STREAMS file contains one of the values shown in Figure 3.

After execution of CONV11	Before execution of CONV14	
	Step converted	Step not converted
10	10	11
11	11	11
12	16	11
13	15	11
14	15	11
15	15	16
16	16	16
17	17	11
18	10	11
19	10	11
2n	2n	2n
3n	3n	5n
4n	4n	5n
5n	5n	5n

11	Not converted
16, 2n, 5n	Ignored
10	Load
15	DOS/VSE format (created and listed)
17	Listed, but not created (statement omitted)

Figure 3. Contents of Positions 1 and 2 of Records of STREAMS File

WORKDSK FILE (DIRECT ACCESS, LENGTH 128)

TYPE 1 = A (CONV00, CONV01, CONV02, CONV07 components)

1	A identifier
2-6	Last relative record
7-107	Logical SYSXXX device used

Type 1 = B (CONV00, CONV01, CONV02, CONV07 components)

1	B identifier
2-30	Label
31-33	Block size or control interval size (packed)
34-36	Record size (packed)
37	File type
38-39	Key length
40-42	Key location (packed)
43-44	Percent free control intervals
45-46	Percent free control areas
47-49	Logical SYSXXX device
50-53	Validity period in days
54-57	Number of records (packed)
58-59	Application initials
60-63	Number of records in prime data area (packed)
64-67	Number of records in free space of control intervals (packed)
68-71	Number of records in independent free space of control area (packed)
72-74	Number of data records in control area (packed)
75-77	Number of index records in control area (packed)
78-80	Number of alternate index records in control area (packed)
81-82	Number of control intervals per control area (packed)
83-84	Number of free control intervals per control area
85-88	Number of data control intervals (packed)
89-92	Number of index control intervals (packed)
93-96	Number of alternate index control intervals (packed)
97-100	Number of data tracks/blocks (packed)
101-104	Number of index tracks/blocks (packed)
105-108	Number of alternate index tracks/blocks (packed)
112	Destination
113-120	System/3 label
125-128	Index control interval size (packed)

LABELS FILE (INDEXED SEQUENTIAL, LENGTH 128)

Type 1 = alphabetic (CONV01, CONV08, CONV09, CONV10, CONV11, components)

1-29	Label
30-32	Block size or control interval size (packed)
33-35	Record length (packed)
36	File type
37-38	Key length
39-41	Key location (packed)
42-43	Percent free control intervals
44-45	Percent free control areas
46-48	Logical SYSXXX device
49-52	Validity period in days
53-56	Number of records (packed)
57-58	Application initials
59-62	Number of records in prime data area (packed)
63-66	Number of records free space of control intervals (packed)
67-70	Number of records in free space of control area (packed)
71-73	Number of data records in control area (packed)
74-76	Number of index records in control area (packed)
77-79	Number of alternate index records in contrcl area (packed)
80-81	Number of control intervals per control area (packed)
82-83	Number of free control intervals per contrcl area
84-87	Number of data control intervals (packed)
88-91	Number of index control intervals (packed)
92-95	Number of alternate index control intervals (packed)
96-99	Number of data tracks/blocks (packed)
100-103	Number of index tracks/blocks (packed)
104-107	Data relative track/block (packed)
111	Destination
112-119	System/3 label
120-125	Pack name

Type 1 = 0 (CONV01, CCNV07, CCNV10 components)

1	0 identifier
2-4	Logical SYSXXX device
32-35	Pack name
36-42	Relative track/block position
43-49	Number of tracks/blocks per data space.

PROGRAM FILE AND CCNV02 WORKDSK FILE (LENGTH 128)

Type 1 = 1 (CONV02, CCNV03, CCNV04, CCNV05, components)

1	1 identifier	
2-46	Relative record number of first statement	
2-6	H statement	(CONV02, CONV03, CONV04, CONV05)
7-11	F statement	
12-16	E statement	
17-21	I statement	
22-26	I statement	(CONV02, CONV03, CONV04, CONV05)
27-31	C statement	
32-36	C-IX statement	
37-41	C-SR statement	
42-46	C statement	
47-96	Relative record number of first statement in table	
47-51	1	
52-56	2	
57-61	3	
62-66	4	
67-71	5	
72-76	6	(CONV02, CONV03, CONV05)
77-81	7	
82-86	8	
87-91	9	
92-96	10	
97-101	Relative record number of /* statement	(CONV02, CONV03, CONV04)
102-106	Relative record number of last record of table 10	(CONV02, CONV03 -- input)
102-103	Free indicator 1	(CONV03 -- output, CONV04)
107-111	Relative record number of last record of source program	(CONV02 -- output, CONV03, CONV04, CONV05)
128	1 if ESDS file accessed via relative record number	

Type 1 = 2 (CONV02, CONV03, CONV04, CONV05 components)

1	2 identifier	
2-121	Names of files used in program	(CONV02, CONV03)
2-9	1	
10-17	2	
18-25	3	
26-33	4	
34-41	5	
42-49	6	
50-57	7	
58-65	8	
66-73	9	
74-81	10	
82-89	11	
90-97	12	
98-105	13	
106-113	14	
114-121	15	
2-66	Errors detected in program	(CONV04, CONV05)
122	Auto Report	(CONV02, CONV04, CONV05)
123-128	Program name	(CONV02, CONV04, CONV05)

Type 1 = 3 (CONV02, CONV03, CONV04, CONV05 components)

1	3 identifier	
2-30	Label	(CONV02)
31-33	Block size or control interval size (packed)	(CONV02, CONV04)
34-36	Record length (packed)	(CONV02, CONV04)
37	File type	(CONV02, CONV04)
38-39	Key length	(CONV02, CONV04)
40-42	Key location (packed)	(CONV02)
43-44	Percent free control intervals	(CONV02)
45-46	Percent free control areas	(CONV02)
47-49	Logical SYSXXX device	(CONV02, CONV04)
104-111	File	(CONV02, CONV03, CONV04)
113-120	System/3 Label	(CONV02, CONV04)

Type 1 = 4 (CONV02, CONV03, CONV04, CONV05 components)

1	4 identifier	
2-81	Source program statement	
82-101	Flagged errors	(CONV02 and CONV03 -- output, CONV04, CONV05)
92-97	Table statement (if present)	(CONV03 -- input)
102	If RAF present	(CONV03, CONV04)
104-105	Record length in F statement	(CONV04)
106-124	Information about E statement	
106-108	Old entry length	(CONV02 and CONV03 -- input)
109-111	Old record length	(CONV02 and CONV03 -- input)
112-114	Old number of records	(CONV02 and CONV03 -- input)
115-117	New number of entries per record	(CONV02 and CONV03 -- input, CONV04)
118-120	New record length	(CONV02 and CONV03 -- input)
121-123	New number of records	(CONV02 and CONV03 -- input)
124	Flag 20	(CONV02 and CONV03 -- input)
124-128	Relative record number or record type 3 in F statement	(CONV03 -- input, CONV04)

Type 1 = 5 (CONV02, CONV03, CONV04 components)

1	5 identifier	
2-81	If blank, source statement added	(CONV02, CONV03, CONV04)
7	If F, F statement for new table file	(CONV02)
16-17	IT	
82-101	Flagged errors	(CONV04)
82-83	1	
84-85	2	
86-87	3	
88-89	4	
90-91	5	
92-93	6	
94-95	7	
96-97	8	
98-99	9	
100-101	10	

Type 1 = 6 (CONV02, CONV03, CONV05 components)

1	6 identifier
2-3	/* -- last source program statement

Type 1 = 7 (CONV02, CONV04 components)

1	7 identifier
2-81	// COPY or Auto Report control statement
82-83	Flag 18

Type 1 = A (CONV06 component)

1	A identifier
2-7	Name of source program
8-9	Sequence number
10-16	File name
17-24	Old file name
25	File type
26-29	Block length
30-33	Record length
34	Processing type
35-42	System/3 label
43-48	System/3 label
43-48	Logical SYSXXX device
49-51	Physical device address
52-56	Relative record number
57-59	Number of FCB phase for print file

Type 1 = B (CONV06 component)

1	B identifier
2-7	Name of source program
8-9	Sequence number
10-12	Forms length
13-15	Line number of channel 12
16-18	Line number of channel 1
19-21	Line number of channel 2
22-24	Line number of channel 3
25-27	Line number of channel 4
28-30	Line number of channel 5
31-33	Line number of channel 6
34-36	Line number of channel 7
37-39	Line number of channel 8
40-42	Line number of channel 9
43-45	Line number of channel 10
46-48	Line number of channel 11
49-51	Line number of channel 12
52-56	Relative record number
57-59	Number of FCB phase

CONTROL FILE (LENGTH 64)

Type 1 = A (CONV04, CONV06 components)

1	A identifier
2-7	Name of source program
8-9	Sequence number
10-16	DOS/VSE file name
17-24	Old System/3 file name
25	File type
26-29	Block length
30-33	Record length
34	Processing type
35-42	System/3 label
43-48	Logical SYSXXX device
49-51	Physical device address
52-56	Relative record number
57-59	Number of first FCB

Type 1 = B (CONV04, CONV06 components)

1	B identifier
2-7	Name of source program
8-9	Sequence number
10-12	Forms length line number
13-15	Line number of channel 12
16-18	Line number of channel 1
19-21	Line number of channel 2
22-24	Line number of channel 3
25-27	Line number of channel 4
28-30	Line number of channel 5
31-33	Line number of channel 6
34-36	Line number of channel 7
37-39	Line number of channel 8
40-42	Line number of channel 9
43-45	Line number of channel 10
46-48	Line number of channel 11
49-51	Line number of channel 12
52-56	Relative record number
57-59	Number of first FCB

OCLFILE FILE (LENGTH 64)

Type 1 = A (CONV06, CONV10 components)

1	A identifier
2-7	Name of source program
8-9	Sequence number
10-16	DOS/VSE file name
17-24	Old System/3 file name
25	File type
26-29	Block length
30-33	Record length
34	Processing type
35-42	System/3 label
43-48	Logical SYSXXX device
49-51	Physical device address
52-56	Relative record number
58-60	Number of FCB phase for print file

S3LABEL FILE (LENGTH 64)

CONV01, CONV08, CONV11 components

1-8	System/3 label
9-37	DOS/VSE label

JCLFILE FILE (INDEXED SEQUENTIAL, LENGTH 128)

Type 1 = 1 (CONV08, CCNV09, CCNV11 components)

1	1 identifier
2-7	Name of program
8-10	Sequence number
11-60	Image of JCL statement
128	* if // EXEC statement

Type 1 = 2 (CONV08, CONV09, CONV11 components)

1	2 identifier
2-7	Name of program
8-10	Sequence number
11-43	Logical SYSXXX devices
11-13	1
14-16	2
17-19	3
20-22	4
23-25	5
26-28	6
29-31	7
32-34	8
35-37	9
38-40	10
41-43	11
44-54	Assign types
44	1
45	2
46	3
47	4
48	5
49	6
50	7
51	8
52	9
53	10
54	11
55-120	Pack names
55-60	1
61-66	2
67-72	3
73-78	4
79-84	5
85-90	6
91-96	7
97-102	8
103-108	9
109-114	10
115-120	11

Type 1 = 3 (CONV08, CONV09, CONV11 components)

1	3 identifier
2-7	Name of program
8-10	Sequence number
11-17	DOS/VSE file name
18-25	Old System/3 file name
26-29	Block length
30-33	Record length
34	File type
35-36	Key length
37-40	Key position
41-48	System/3 label
49-77	DOS/VSE label
78-81	Auxiliary label
82-85	Validity period
86-88	Logical SYSXXX device
89-94	Pack name
95-98	File type
99-105	Data tracks/blocks
106-112	Data relative track

STREAMS FILE (LENGTH 128, CONV11, CONV12, CONV13, CONV14 COMPONENTS)

Type 1 = 1 (Image of source program statement)

1	1 identifier
2	Statement type: 0 - LCAD (except for Sort and Copy programs) 1 - Not converted 2 - System/3 provisional (to be deleted or printed as not converted) 3 - System/3 conversion pending 4 - Common to System/3 and DOS/VSE 5 - System/3 format 6 - Deleted 7 - To be printed as ignored 8 - LCAD (for Sort programs) 9 - LCAD (for Copy programs)
3-98	Image of source program statement
99	Blank
100	Skip (1 for LOAD and CALL statements)
101	EXEC (* for each program to be executed)
102-107	Program name
108	Number of statement not converted (PAUSE, DATE, SWITCH, JOB)
109-110	For free format, position of the first character of parameter
123	Previous step not converted
124	This step not converted
125-126	Previous step error
127-128	This step error

Type 1 = 2 (Separator between OCI Statements and Sort and Copy control statements)

1	1 identifier
2	Program: 1 - Sort program 2 - Copy program

Type 1 = 3 (ASSGN statements)

1	3 identifier
2-37	Array of up to 12 logical SYSXXX devices
38-109	Array of up to 12 physical device addresses
110-121	Array of up to 12 device types (U = unit record, T = magnetic tape, blank = disk)

Type 1 = 4 (DLBI and TIBL statements)

1	4 identifier
2-8	File name
9-37	Label
38-41	Auxiliary label (path or alternate index)
42-45	Validity period in days
46-49	File type (SD, SVL, VSAM)
50	Type (S, V, T, K, E, R, A)
51-53	Logical SYSXXX device
54-59	VOLID of pack or tape
60-66	Data relative track
67-73	Number of data tracks
74-78	Block length
79-83	Record length
84-85	Key length
86-90	Key location

Type 1 = 5 (Additional record for future use)

1	5 identifier

APPENDIX C. EXECUTION OF SAMPLE PROBLEM

```
* * * * CARDS USED IN THE CUSTOMIZATION * * * *

// LOAD BSCOPY,F1
// RUN
// CLEAR FROM-D1B,PACK-D1D1D1,AREA-CONVER,TYPE-FORCE
// END
// LOAD BDCOPY,F1
// FILE NAME-BACKUP,UNIT-T1,REEL-NL
// RUN
// COPYPACK TO-R1
// END
// LOAD CST151,R1
// SWITCH 10001000
// FILE NAME-CARDIN,UNIT-R1,PACK-CONVER,TRACKS-2
// RUN
CNVMECU1  DISK40 TAPE    DISKET 01000
FUTFBA 4844Y3203-5
EQUDISK45 FBA
EQUTAPE   TAPE        000280
EQUPRINTERPRINTER 132LST00E
EQUMECU1  READ01      IPT00C
EQUMECU2  FBA
EQUMECU2  DISKET    9600100A
/*
// LOAD CST152,R1
// FILE NAME-CARDIN,UNIT-R1,PACK-CONVER,RETAIN-S
// FILE NAME-SOURCE,UNIT-R1,PACK-CONVER
// RUN



* * * * INPUT SAMPLE CARDS * * * *

// LOAD CONV00,R1
// FILE NAME-WORKDSK,UNIT-D1,PACK-D1D1D1,LOCATION-20,TRACKS-30
// RUN
FSAM FILE 1                          20S          002  30  1000SMFFILE1
FKSDS.FILE2                          30K05   11515003  60  2000SMUFILE2
FESDS.FILE3                          20E          003  90  3000SMNFILE3
FSAM FILE 4                          20S          002 120  4000SMNFILE4
/*
```

```
// LOAD CONV01,R1
// FILE NAME-WORKDSK,UNIT-D1,PACK-D1D1D1,LOCATION-20,TRACKS-30
// FILE NAME-WORKDS1,UNIT-D1,PACK-D1D1D1,LOCATION-20,TRACKS-30,LABEL-WORKDSK
// FILE NAME-S3LABEL,UNIT-D1,PACK-D1D1D1,LOCATION-22,TRACKS-10
// FILE  NAME-LABELS,UNIT-D1,PACK-D1D1D1,LOCATION-21 10,TRACKS-10
// RUN
// LOAD CONV02,R1
// FILE NAME-WORKDSK,UNIT-D1,PACK-D1D1D1,LOCATION-20,TRACKS-30
// FILE NAME-PROGRAM,UNIT-D1,PACK-D1D1D1,LOCATION-22 10,TRACKS-50
// RUN
FILE1   SAM FILE 1
FILE2   KSDS.FILE2
//
     H   C  16       J                                                                    SAMPL 1
     FCARD    IP      96  96             MFCU1
     FFILE1   O      240  20             DISK45
     FFILE2   O      210  30 05AI     1 DISK45
     ICARD    NS  01   SNC
     I                                          1   50LAST
     C   01                  SETON                  LR
     CLR           LRTAG1    TAG
     CLR           NUMB      COMP LAST                      02
     CLR 02                  GOTO LRTAG2
     CLR           NUMB      ADD  1          NUMB    50
     CLR                     SETON                  03
     CLR                     EXCPT
     CLR                     GOTO LRTAG1
     CLR           LRTAG2    TAG
     OFILE1   E         03
     O                                      15 'FILE 1 - RECORD'
     O                           NUMB   Z   20
     OFILE2   E         03
     O                           NUMB   Z    5
     O                           NUMB   Z   30
     O                                      25 'FILE 2 - RECORD'
/*
// LOAD CONV02,R1
// FILE NAME-WORKDSK,UNIT-D1,PACK-D1D1D1,LOCATION-20,TRACKS-30
// FILE NAME-PROGRAM,UNIT-D1,PACK-D1D1D1,LOCATION-22 10,TRACKS-50
// RUN
FILE1   SAM FILE 1
FILE2   KSDS.FILE2
```

SAMPL2

```
     H   C  16       J
     FFILE1   IP     240  20              DISK45
     FFILE2   C      240  30R05AI      1 DISK45
     FLIST    O      120 120      OF      PRINTER
     IFILE1   NS  02
     I                                          1  20 TEX1
     I                                         16  20ONUMB
     IFILE2   NS  02
     I                                         11  30 TEX2
     C   01       NUMB       CHAINFILE2                03
     OLIST    O  30&    1P
     O        OR         OF
     O                                    35 'HEADER'
     O         D         01
     O                           TEX1       25
     O                           TEX2       55
     O         T      60 LR
     O                                    34 'END'
/*
// LOAD CONV02,P1
// FILE NAME-WORKDSK,UNIT-D1,PACK-D1D1D1,LOCATION-20,TRACKS-30
// FILE NAME-PROGRAM,UNIT-D1,PACK-D1D1D1,LOCATION-22 10,TRACKS-50
// RUN
FILE1   FSDS,FILE3
```

SAMPL3

```
     H   C  16       J
     ICARD    IP      96  96              MFCU1
     FFILE1   IC     240  20R05AI      1 DISK45
     FLIST    O      120 120      OF      PRINTER
     ICARD    NS  01
     I                                        1   50NUM
     IFILE1   NS  02
     I                                        6  20 TEXT
     C            NUM        CHAINFILE1                03
     OLIST    O  1      01 02N03
     O                           NUM    Z     5
     O                           TEXT       21
/*
// LOAD CONV03,P1
// FILE NAME-PROGRAM,UNIT-D1,PACK-D1D1D1
// FILE   NAME-PROGR,UNIT-D1,PACK-D1D1D1,LABEL-PROGRAM
```

```
// RUN
// LOAD CONV04,R1
// FILE NAME-PROGRAM,UNIT-D1,PACK-D1D1D1,TRACKS-50,LOCATION-22 10
// FILE NAME-CONTROL,UNIT-D1,PACK-D1D1D1,TRACKS-10,LOCATION-25
// RUN
001
/*
// LOAD CONV05,R1
// FILE NAME-PROGRAM,UNIT-D1,PACK-D1D1D1
// RUN
// LOAD CONV06,R1
// FILE NAME-CONTROL,UNIT-D1,PACK-D1D1D1,RETAIN-S
// FILE NAME-PROGRAM,UNIT-D1,PACK-D1D1D1,RETAIN-S
// FILE NAME-OCLFILE,UNIT-D1,PACK-D1D1D1,TRACKS-10,LOCATION-25 10
// RUN
// LOAD CONV07,R1
// FILE NAME-WORKDSK,UNIT-D1,PACK-D1D1D1,RETAIN-S
// FILE  NAME-LABELS,UNIT-D1,PACK-D1D1D1
// FILE NAME-LABELS1,UNIT-D1,PACK-D1D1D1,LABEL-LABELS
// RUN
002VOL001     352
003VOL002     700
/*
// LOAD CONV08,R1
// FILE NAME-OCLFILE,UNIT-D1,PACK-D1D1D1
// FILE NAME-S3LABEL,UNIT-D1,PACK-D1D1D1
// FILE  NAME-LABELS,UNIT-D1,PACK-D1D1D1
// FILE NAME-JCLFILE,UNIT-D1,PACK-D1D1D1,TRACKS-20,LOCATION-26
// RUN
// LOAD CONV09,R1
// FILE NAME-JCLFILE,UNIT-D1,PACK-D1D1D1
// FILE  NAME-LABELS,UNIT-D1,PACK-D1D1D1
// RUN
// LOAD CONV10,R1
// FILE  NAME-LABELS,UNIT-D1,PACK-D1D1D1
// FILE NAME-LABELS1,UNIT-D1,PACK-D1D1D1,LABEL-LABELS
// RUN
// LOAD CONV11,R1
// FILE NAME-LABELS,UNIT-D1,PACK-D1D1D1
// FILE NAME-JCLFILE,UNIT-D1,PACK-D1D1D1
// FILE NAME-S3LABEL,UNIT-D1,PACK-D1D1D1
// FILE NAME-STREAMS,UNIT-D1,PACK-D1D1D1,TRACKS-20,LOCATION-027 00
```

```
// RUN
// LOAD SAMPL1,R1
// SWITCH 00010000
// DATE 101278
// FILE NAME-FILE2,UNIT-D1,PACK-D1D1D1,TRACKS-10
// FILE NAME-FILE1,UNIT-D1,PACK-D1D1D1,TRACKS-10,LABEL-FILE4
// RUN
// LOAD RDSORT,R1
// FILE   NAME-INPUT,UNIT-D1,PACK-D1D1D1,TRACKS-10,LABEL-FILE1
// FILE  NAME-OUTPUT,UNIT-D1,PACK-D1D1D1,TRACKS-10,LABEL-FILE3
// RUN
     HSORTP      5A        S X  20
ALTSEQ  EACB
**
     I C    1    2NEC
     END    1    5
     EDC    1    5
     EDC    6   15
     EDC   16   20
// END
// LOAD SAMPL2,R1
// FILE NAME-FILE1,UNIT-D1,PACK-D1D1D1,TRACKS-10,LABEL-FILE4
// FILE NAME-FILE2,UNIT-D1,PACK-D1D1D1,TRACKS-10
// RUN
// LOAD RDSORT,R1
// FILE NAME-INPUT,UNIT-D1,PACK-D1D1D1,LABEL-FILE4
// FILE NAME-OUTPUT,UNIT-D1,PACK-D1D1D1,TRACKS-10,LABEL-FILE1
// RUN
     HSORTP      5A          X  20
     END    1    5
     EDC    1   20
// END
// LOAD RCOPY,R1
// FILE NAME-COPYIN,UNIT-D1,PACK-D1D1D1,LABEL-FILE3
// RUN
// COPYFILE OUTPUT-PRINT
// SELECT KEY,FROM-'00012'
// END
// LOAD SAMPL3,R1
// FILE NAME-FILE1,UNIT-D1,PACK-D1D1D1,LABEL-FILE3
// RUN
// LOAD RCOPY,R1
```

```
// FILE NAME-COPYIN,UNIT-F1,PACK-FIF1F1,TRACKS-10,LABEL-FILE2
// RUN
// COPYFILE OUTPTX-PRINT
// SELECT KEY,FROM-'00012'
// END
/*
// LOAD CONV12,R1
// FILE NAME-STREAMS,UNIT-D1,PACK-D1D1D1
// FILE NAME-STREAM1,UNIT-D1,PACK-D1D1D1,LABEL-STREAMS
// RUN
// LOAD CONV13,R1
// FILE NAME-STREAMS,UNIT-D1,PACK-D1D1D1
// FILE NAME-STREAM1,UNIT-D1,PACK-D1D1D1,LABEL-STREAMS
// RUN
// LOAD CONV14,R1
// FILE NAME-STREAMS,UNIT-D1,PACK-D1D1D1
// FILE NAME-STREAM1,UNIT-D1,PACK-D1D1D1,LABEL-STREAMS
// RUN



* * * * OUTPUT SAMPLE CARDS * * * *

* FE JOB JNM=SAMPL1,CLASS=0
// JOB SAMPL1 COMPILE AND LNKEDT
// OPTION CATAL
   PHASE SAMPL1,S
// EXEC FPGII,SIZE=64K
01010H                J                                                    SAMPL1
02010FCARD    IP  F       80            READ01 SYSIPT                      SAMPL1
02020FFILE1   O   F1000   20            FCA    SYS002S                     SAMPL1
02030FFILE2   O   F       30            KSDS                               SAMPL1
03010ICARD    NS  01    5NC                                                SAMPL1
03020I                                        1    50LAST                  SAMPL1
04010C    01                  SETLN             LR                        SAMPL1
04020CLR          LRTAG1      TAG                                         SAMPL1
04030CLR          NUMB        COMP LAST                  02               SAMPL1
04040CLR 02                   GETC LRTAG2                                 SAMPL1
04050CLR          NUMB        ADD  1          NUMB   50                  SAMPL1
04060CLR                      SETON                 03                    SAMPL1
04070CLR                      EXCPT                                       SAMPL1
```

```
04080CLR                        GOTO LRTAG1                                     SAMPL 1
04090CLR            LRTAG2      TAG                                             SAMPL 1
050100FILE1    F            03                                                  SAMPL 1
050200                                          15 'FILE 1 - RECORD'            SAMPL 1
050300                              NUMB   Z    20                             SAMPL 1
050400FILE2    F            03                                                  SAMPL 1
050500                              NUMB   Z     5                             SAMPL 1
050600                              NUMB   Z    30                             SAMPL 1
050700                                          25 'FILE 2 - RECORD'            SAMPL 1
/*
// EXEC LNKEDT,SIZE=64K
/&
* $$ EOJ
* $$ JOB JNM=SAMPL2,CLASS=0
// JOB SAMPL2 COMPILE AND LNKEDT
// OPTION CATAL
   PHASE SAMPL2,S
// EXEC RPGII,SIZE=64K
01010H                  J                                                       SAMPL 2
02010FFILE1    IP   F1000  20             FBA     SYS002S                      SAMPL 2
02020FFILE2    C    F       300 5A        KSDS                                 SAMPL 2
02030FLIST     O    F      132      OF     PRINTERSYSLST                        SAMPL 2
03010L*  066FL00601                                                           06012SAMPL 2
04010IFILE1    NS  02                                                          SAMPL 2
040201                                             1   20 TEX1                 SAMPL 2
040301                                            16  200NUMB                  SAMPL 2
04040IFILE2    NS  02                                                          SAMPL 2
040501                                            11   30 TEX2                 SAMPL 2
05010C   01        NUMB       CHAINFILE2                 03                   SAMPL 2
06010FLIST     D  301    1P                                                    SAMPL 2
06020F         OR         OF                                                   SAMPL 2
060300                                          35 'HEADER'                     SAMPL 2
06040P          D  1      01                                                   SAMPL 2
060500                              TEX1       25                             SAMPL 2
060600                              TEX2       55                             SAMPL 2
06070F          T      12 LR                                                   SAMPL 2
060800                                          34 'END'                        SAMPL 2
/*
// EXEC LNKEDT,SIZE=64K
/&
* $$ EOJ
* $$ JOB JNM=EXECB001,CLASS=0
```

```
// JOB FBFCB001 ASSEMBLY AND LINKEDIT
// OPTION CATAL
   PHASE FBFCB001,+0
// EXEC ASSEMBLY,SIZE=64K
FBFCB001 CSECT
         DC    X'00000000001C0000000000000000C0000000000000000000000000'
         DC    X'C0000000000000000000000000000000000000000000000C0000'
         DC    X'0000000C0C00000C0000000000100C0000000000000000000000000'
         DC    X'C00C0000000000000000C0C0C0C0000C000C0000000000000000000'
         DC    X'0000000000000000',XL80'00',CL5' '
         DC    CL75'FBFCB001 LOADED'
         END
/*
// EXEC LNKEDT,SIZE=64K
/&
* $$ EOJ
* TAP001 TO COPY S 3 'FILE1' TO NEW 'SAM FILE 1' TYPE S
// LOAD $TINIT,F1
// RUN
// VOL UNIT-T1,TYPE-CLEAR,REEL-TAP001,DENSITY-1600
// END
// LOAD $COPY,F1
// FILE NAME-COPYIN,LABEL-FILE1,UNIT-   ,PACK-
// FILE NAME-COPYO,LABEL-FILE1,UNIT-T1,
// REEL-TAP001,BLKL-4080,RECL-20,RECFM-FB
// RUN
// COPY FILE OUTPUT-FILE
// END
* * * DEFINE M.CATAL. MASTER.CATALOG * * *
// DLBL IJSYSCT,'MASTER.CATALOG',9999,VSAM
// EXTENT SYSCAT,      ,,,      ,256
// EXEC IDCAMS,SIZE=AUTO
   DEFINE       MCAT-
                (NAME(MASTER.CATALOG)-
                FILE(IJSYSCT)-
                BLOCKS(256)-
                VOLUME(       ))
/*
* * * DEFINE SPACE VVOL002.DATA.SPACE.L0000000.L.SPAC * * *
// ASSGN SYS003,FBA,VOL=VOL002,SHR
// DLBL DATASP,'VVOL002.DATA.SPACE.L0000000.L.SPAC',9999,VSAM
// EXTENT SYS003,VOL002,,,704,320
```

```
// EXEC IDCAMS,SIZE=AUTO
   DEFINE       SPACE-
                (BLOCKS(320)-
                FILE(DATASP)-
                VOLUMES(VOL002))-
                CATALOG(MASTER.CATALOG)
/*
* * * DEFINE ESDS ESDS.FILE3 * * *
// EXEC IDCAMS,SIZE=AUTO
   DEFINE       CLUSTER-
                (NAME(ESDS.FILE3)-
                VOLUMES(VOL002)-
                FOR(90)-
                RECSZ(20 20)-
                CISZ(1024)-
                SPEED-
                RECORDS(3200 800)-
                SHR(2)-
                REUSE-
                NIXD)-
                DATA-
                (NAME(ESDS.FILE3.L0000000.LCCCCCCC.DATA))-
                CATALOG(MASTER.CATALOG)
/*
* * * DEFINE KSDS KSDS.FILE2 * * *
// EXEC IDCAMS,SIZE=AUTO
   DEFINE       CLUSTER-
                (NAME(KSDS.FILE2)-
                VOLUMES(VOL002)-
                FOR(60)-
                RECSZ(30 30)-
                SPEED-
                SHR(2)-
                FSPC(12 13)-
                KEYS(5 0))-
                DATA-
                (NAME(KSDS.FILE2.L0000000.LCCCCCCC.DATA)-
                RECORDS(2640 406)-
                CISZ(1024))-
                INDEX-
                (NAME(KSDS.FILE2.L0000000.L0000000.INDX)-
                CISZ(512))-
```

```
                CATALOG(MASTER.CATALOG)
/*
* * * DATA IMPORT UTILITY TO COPY FILE KSDS.FILE2 * * *
// ASSGN SYS004,3340,VOL=
// ASSGN SYS007,FBA,VOL=VOL002,SHR
// DLBL OUTPUT,'KSDS.FILE2',60,VSAM
// EXTENT SYS007,VOL002
// EXEC S3DATA,SIZE=AUTO
// FILE NAME-FILE2,PACK-
// OUTP DLBL-OUTPUT,AM-VSAM
/*
* * * REPRO TO COPY FILE SAM FILE 1 * * *
// ASSGN SYS004,TAPE,VOL=TAP001
// ASSGN SYS005,FBA,VOL=VOL001,SHR
// TLBL TAPEIN,'FILE1',,TAP001
// DLBL FILEOUT,'SAM FILE 1',30,SD
// EXTENT SYS005,VOL001,,,352,64
// EXEC IDCAMS,SIZE=AUTO
   REPRO        INFILE(TAPEIN-
                ENV-
                (RECFM(FB)-
                BLKSZ(4080)-
                RECSZ(20)-
                DDEV(2400)))-
                OUTFILE(FILEOUT-
                ENV-
                (RECFM(FB)-
                BLKSZ(1000)-
                RECSZ(20)-
                DDEV(FBA)))
/*
* * * SAMPL1 EXECUTION * * *
// UPSI 00010000
// DATE 10/12/78
// ASSGN SYSIPT,X'00C'
// ASSGN SYS002,FBA,VOL=VOL001,SHR
// ASSGN SYS003,FBA,VOL=VOL002,SHR
// DLBL FILE1,'SAM FILE 4',120,SD
// EXTENT SYS002,VOL001,,,416,192
// DLBL FILE2,'KSDS.FILE2',60,VSAM
// EXTENT SYS003,VOL002
// EXEC SAMPL1,SIZE=AUTO
```

```
* * * SORT   EXECUTION * * *
// ASSGN SYS002,FBA,VOL=VOL002,SHR
// ASSGN SYS003,FBA,VOL=VOL001,SHR
// DLBL SORTOUT,'ESDS.FILE3',90,VSAM
// EXTENT SYS002,VOL002
// DLBL SORTIN1,'SAM FILE 1',30,SD
// EXTENT SYS003,VOL001,,,352,64
// EXEC SORT,SIZE=64K
   OPTION PRINT=ALL,SORTWK=001,SORTOUT=002,SORTIN=003
   ALTSEQ CODE=(FAFB)
   INCLUDE COND=(1,2,AQ,NE,C' ')
   SORT FIELDS=(1,5,ZD,A),FILES=1,WORK=1
   RECORD TYPE=F,LENGTH=(20)
   INPFIL BLKSIZE=1000
   OUTFIL ESDS,REUSE
   END
/*
* * * SAMPL2 EXECUTION * * *
// ASSGN SYSLST,X'00E'
* DLBL LST FCB=FBFCB001,LST=SYSLST
// ASSGN SYS002,FBA,VOL=VOL001,SHR
// ASSGN SYS003,FBA,VOL=VOL002,SHR
// DLBL FILE1,'SAM FILE 4',120,SD
// EXTENT SYS002,VOL001,,,416,192
// DLBL FILE2,'KSDS.FILE2',60,VSAM
// EXTENT SYS003,VOL002
// EXEC SAMPL2,SIZE=AUTO
* * * COPY   EXECUTION * * *
// ASSGN SYS003,FBA,VOL=VOL002,SHR
// DLBL INFILE,'KSDS.FILE2',60,VSAM
// EXTENT SYS003,VOL002
// EXEC IDCAMS,SIZE=AUTO
   PRINT        INFILE(INFILE) DUMP-
                FKEY(00012)
/*
```

LABEL	TYPE	PRIMARY RECORDS	RECORDS IN C. I. FREE SPACE	RECORDS IN C. A. FREE SPACE	BLOCKS IN DATA COMPONENT	BLOCKS IN INDEX DATASET	BLOCKS IN ALTER. INDEX	TOTAL BLOCKS	SYSTEM 3 LABEL
SAM FILE 1	S	1.550			64			64	FILE1
SAM FILE 4	S	4.750			192			192	FILE4
TOTAL NO. OF BLOCKS SYS002								256	

LABEL	TYPE	PRIMARY RECORDS	RECORDS IN C. I. FREE SPACE	RECORDS IN C. A. FREE SPACE	BLOCKS IN DATA COMPONENT	BLOCKS IN INDEX DATASET	BLOCKS IN ALTER. INDEX	TOTAL BLOCKS	SYSTEM 3 LABEL
ESDS.FILE3	E	3.200			128			128	FILE3
KSDS.FILE2	K	2.030	280	330	160	32		192	FILE2
TOTAL NO. OF BLOCKS SYS003								320	
TOTAL NO. OF BLOCKS APLICATION SM								576	

LOGICAL SYS USED

002 003

LISTING OF ERRORS DETECTED

```
FFILE1   IC    240  20R05AI      1 DISK45

                    FILE ATTRIBUTES DO NOT MATCH
     ALLOWED - TYPES KA    - RECORD   20 - KLENG  5 - KLOC    1
     SUBMITED - TYPE E     - RECORD   20 - KLENG    - KLOC
```

ERRORS DETECTED IN SOURCE PROGRAM SAMPL3 - CORRECT IT, AND REPEAT THE STEP.

09 NON IDENTIFIED RECORD CAN CANCEL THE PROGRAM.- MAY BE NECESSARY TO CHECK THE *ERROR FIELD.

```
* EP JOB JNM=SAMPL1,CLASS=0
// JOP SAMPL1 COMPILE AND LNKEDT
// OPTION CATAL
   PHASE SAMPL1,S
// EXEC RPGII,SIZE=64K


01010H             J                                                    SAMPL1


02010FCARD    IP  F       80            READ01 SYSIPT                  SAMPL1 *09
02020FFILE1   O   F1000   20            FBA    SYS002S                 SAMPL1
02030FFILE2   O   F       30            KSDS                           SAMPL1


03010ICARD    NS  01    5NC                                            SAMPL1
03020I                                      1    50LAST                SAMPL1


04010C   01                 SETON                    LR                SAMPL1
04020CLR          LRTAG1    TAG                                        SAMPL1
04030CLR          NUMB      COMP LAST                     02           SAMPL1
04040CLR 02                 GOTO LRTAG2                                SAMPL1
04050CLR          NUMB      ADD  1          NUMB     50                SAMPL1
04060CLR                    SETON                 03                   SAMPL1
04070CLR                    EXCPT                                      SAMPL1
04080CLR                    GOTO LRTAG1                                SAMPL1
04090CLR          LRTAG2    TAG                                        SAMPL1


05010OFILE1   E         03                                             SAMPL1
```

```
050200                                          15 'FILE 1 - RECORD'          SAMPL1
050300                          NUMB  Z         20                            SAMPL1
050400FILE2    F         03                                                   SAMPL1
050500                          NUMB  Z          5                            SAMPL1
050600                          NUMB  Z         30                            SAMPL1
050700                                          25 'FILE 2 - RECORD'          SAMPL1
/*
// EXEC LNKEDT,SIZE=64K
/&
* $$ EOJ
```

WARNING MESSAGES FOR THE PROGRAM SAMPL2

46 L COMMENT CARD HAS BEEN ADDED.

61 CHANGED SKIP TO A LINE NUMBER BY SKIP TO A CHANNEL.

62 ONE SPACE AFTER ASSUMED.

```
* ** JOB JNM=SAMPL2,CLASS=0
// JOB SAMPL2 COMPILE AND LNKEDT
// OPTION CATAL
   PHASE SAMPL2,S
// EXEC RPGII,SIZE=64K


01010H             J                                                    SAMPL2


02010FFILE1   IP  F1000  20            FBA     SYS002S                 SAMPL2
02020FFILE2   C   F       30R 5A       KSDS                            SAMPL2
02030FLIST    O   F     132      OF    PRINTERSYSLST                   SAMPL2


03010I*  066FL00601                                                    06012SAMPL2   4


04010IFILE1   NS  02                                                   SAMPL2
04020I                                    1  20 TEX1                   SAMPL2
04030I                                   16  200NUMB                  SAMPL2
04040IFILE2   NS  02                                                   SAMPL2
04050I                                   11  30 TEX2                   SAMPL2


05010C   01       NUMB      CHAINFILE2              03                 SAMPL2


06010OLIST    O  301   1P                                              SAMPL2   61
```

```
060200         OR          OF                                                  SAMPL2
060300                                          35 'HEADER'                    SAMPL2
060400           D  1        01                                                SAMPL2  62
060500                                  TEX1    25                             SAMPL2
060600                                  TEX2    55                             SAMPL2
060700           T       12 LR                                                 SAMPL2  61
060800                                          34 'END'                       SAMPL2
/*
// EXEC LNKEDT,SIZE=64K
/&
* PR EOJ
```

CARRIAGE CONTROL TAPE 001 OR FCB FRFCB001

CHANNELS	FL	01	02	03	04	05	06	07	08	09	10	11	12
LINES	66	6											60

USED BY PROGRAMS

SAMPL2

LABEL	TYPE	SYS LOG.	BLOCKS IN DATA COMPO.	BLOCKS IN INDEX D.SET	BLOCKS IN ALTER.INDEX	TOTAL BLOCKS
SAM FILE 1	S	002	64			64
SAM FILE 4	S	002	192			192
				TOTAL NO. OF BLOCKS ON PACK VOL001		256

L A B E L	TYPE	SYS LOG.	BLOCKS IN DATA COMPO.	BLOCKS IN INDEX D.SET	BLOCKS IN ALTER.INDEX	TOTAL BLOCKS
ESDS.FILE3	E	003	128			128
KSDS.FILE2	K	003	160	32		192
TOTAL NO. OF BLOCKS ON PACK VOL002						320

```
// DLBL          ,'ESDS.FILE3',90,VSAM
// EXTENT SYS003,VOL002

// DLBL          ,'KSDS.FILE2',60,VSAM
// EXTENT SYS003,VOL002

// DLBL          ,'SAM FILE 1',30,SD
// EXTENT SYS002,VOL001,1,1,352,64

// DLBL          ,'SAM FILE 4',120,SD
// EXTENT SYS002,VOL001,1,1,416,192
```

********** SAMPL1 **********

```
// ASSGN SYSIPT,X'00C'
// ASSGN SYS002,FBA,VOL=VOL001,SHR
// ASSGN SYS003,FBA,VOL=VOL002,SHR
// DLBL FILE1,'SAM FILE 1',30,SD
// EXTENT SYS002,VOL001,1,1,352,64
// DLBL FILE2,'KSDS.FILE2',60,VSAM
// EXTENT SYS003,VOL002
// EXEC SAMPL1,SIZE=AUTO
```

********** SAMPL2 **********

```
// ASSGN SYSLST,X'00E'
* // DLBL LST FCB=FBFCB001,LST=SYSLST
// ASSGN SYS002,FBA,VOL=VOL001,SHR
// ASSGN SYS003,FBA,VOL=VOL002,SHR
// DLBL FILE1,'SAM FILE 1',30,SD
// EXTENT SYS002,VOL001,1,1,352,64
// DLBL FILE2,'KSDS.FILE2',60,VSAM
// EXTENT SYS003,VOL002
// EXEC SAMPL2,SIZE=AUTO
```

```
* TAP001 TO COPY S 3 'FILE1' TO NEW 'SAM FILE 1' TYPE S

// LOAD RTINIT,F1
// RUN
// VOL UNIT-T1,TYPE-CLEAR,REEL-TAP001,DENSITY-1600
// END
// LOAD RCOPY,F1
// FILE NAME-COPYIN,LABEL-FILE1,UNIT-   ,PACK-
// FILE NAME-COPYO,LABEL-FILE1,UNIT-T1,
// REEL-TAP001,BLKL-4080,RECL-20,RECFM-FB
// RUN
// COPY FILE OUTPUT-FILE
// END
```

```
* * * DEFINE M.CATAL. MASTER.CATALOG * * *
// DLBL IJSYSCT,'MASTER.CATALOG',9999,VSAM
// EXTENT SYSCAT,      ,,,      ,256
// EXEC IDCAMS,SIZE=AUTO
   DEFINE       MCAT-
                (NAME(MASTER.CATALOG)-
                FILE(IJSYSCT)-
                BLOCKS(256)-
                VOLUME(      ))
/*


* * * DEFINE SPACE VVOL002.DATA.SPACE.L0000000.L.SPAC * * *
// ASSGN SYS003,DBA,VOL=VOL002,SHR
// DLBL DATASP,'VVOL002.DATA.SPACE.L0000000.L.SPAC',9999,VSAM
// EXTENT SYS003,VOL002,,,704,320
// EXEC IDCAMS,SIZE=AUTO
   DEFINE       SPACE-
                (BLOCKS(320)-
                FILE(DATASP)-
                VOLUMES(VOL002))-
                CATALOG(MASTER.CATALOG)
/*


* * * DEFINE ESDS ESDS.FILE3 * * *
// EXEC IDCAMS,SIZE=AUTO
   DEFINE       CLUSTER-
                (NAME(ESDS.FILE3)-
                VOLUMES(VOL002)-
                FOR(90)-
                RECSZ(20 20)-
                CISZ(1024)-
                SPEED-
```

```
                RECORDS(3200 800)-
                SHR(2)-
                REUSE-
                NIXD)-
                DATA-
                (NAME(ESDS.FILE3.L0000000.L0000000.DATA))-
                CATALOG(MASTER.CATALOG)
/*


* * * DEFINE KSDS KSDS.FILE2 * * *
// EXEC IDCAMS,SIZE=AUTO
   DEFINE       CLUSTER-
                (NAME(KSDS.FILE2)-
                VOLUMES(VOL002)-
                FOR(60)-
                RECSZ(30 30)-
                SPEED-
                SHR(2)-
                FSPC(12 13)-
                KEYS(5 0))-
                DATA-
                (NAME(KSDS.FILE2.L0000000.L0000000.DATA)-
                RECORDS(2640 406)-
                CISZ(1024))-
                INDEX-
                (NAME(KSDS.FILE2.L0000000.L0000000.INDX)-
                CIS7(512))-
                CATALOG(MASTER.CATALOG)
/*
* * * DATA IMPORT UTILITY TO COPY FILE KSDS.FILE2 * * *
// ASSGN SYS004,3340,VOL=
// ASSGN SYS007,FBA,VOL=VOL002,SHR
// DLBL OUTPUT,'KSDS.FILE2',60,VSAM
// EXTENT SYS007,VOL002
```

```
// EXEC S3DATA,SIZE=AUTO
// FILE NAME-FILE2,PACK-
// OUTP DLBL-OUTPUT,AM-VSAM
/*


* * * REPRO TO COPY FILE SAM FILE 1 * * *
// ASSGN SYS004,TAPE,VOL=TAP001
// ASSGN SYS005,FBA,VOL=VOL001,SHR
// TLBL TAPEIN,'FILE1',,TAP001
// DLBL FILEOUT,'SAM FILE 1',30,SD
// EXTENT SYS005,VOL001,,,352,64
// EXEC IDCAMS,SIZE=AUTO
   REPRO        INFILE(TAPEIN-
                ENV-
                (RECFM(FB)-
                BLKSZ(4080)-
                RECSZ(20)-
                RDEV(2400)))-
                OUTFILE(FILEOUT-
                ENV-
                (RECFM(FB)-
                BLKSZ(1000)-
                RECSZ(20)-
                RDEV(FBA)))
/*
```

3 NO INFORMATION ABOUT THIS PROGRAM.

28 DIGIT PORTION USED (0 IN COL.8)

46 SELECT KEY WITH UNKEYED FILE.

```
* * * SAMPL1 EXECUTION * * *
// UPSI 00010000
// DATE 10/12/78
// ASSGN SYSIPT,X'00C'
// ASSGN SYS002,FBA,VOL=VOL001,SHR
// ASSGN SYS003,FBA,VOL=VOL002,SHR
// DLBL FILE1,'SAM FILE 4',120,SD
// EXTENT SYS002,VOL001,,,416,192
// DLBL FILE2,'KSDS.FILE2',60,VSAM
// EXTENT SYS003,VOL002
// EXEC SAMPL1,SIZE=AUTO
            * * * * INSERT ANY DATA HERE * * * *


* * * SORT   EXECUTION * * *
// ASSGN SYS002,FBA,VOL=VOL002,SHR
// ASSGN SYS003,FBA,VOL=VOL001,SHR
// DLBL SORTOUT,'ESDS.FILE3',90,VSAM
// EXTENT SYS002,VOL002
// DLBL SORTIN1,'SAM FILE 1',30,SD
// EXTENT SYS003,VOL001,,,352,64
// EXEC SORT,SIZE=64K
   OPTION PRINT=ALL,SORTWK=001,SORTOUT=002,SORTIN=003
   ALTSEQ CODE=(FAFB)
   INCLUDE COND=(1,2,AQ,NE,C' ')
   SORT FIELDS=(1,5,70,A),FILES=1,WORK=1
   RECORD TYPE=F,LENGTH=(20)
   INPFIL BLKSIZE=1000
   OUTFIL ESDS,REUSE
   END
/*


* * * SAMPL2 EXECUTION * * *
```

```
// ASSGN SYSLST,X'00E'
* $$ LST FCB=FBFCB001,LST=SYSLST
// ASSGN SYS002,FBA,VOL=VOL001,SHR
// ASSGN SYS003,FBA,VOL=VOL002,SHR
// DLBL FILE1,'SAM FILE 4',120,SD
// EXTENT SYS002,VOL001,,,416,192
// DLBL FILE2,'KSDS.FILE2',60,VSAM
// EXTENT SYS003,VOL002
// EXEC SAMPL2,SIZE=AUTO
          * * * * INSERT ANY DATA HERE * * * *


// LOAD $DSORT,R1                                                    * NOT CONVERTED *
// FILE NAME-INPUT,UNIT-D1,PACK-D1D1D1,LABEL-FILE4                   * NOT CONVERTED *
// FILE NAME-OUTPUT,UNIT-D1,PACK-D1D1D1,TRACKS-10,LABEL-FILE1        * NOT CONVERTED *
// RUN                                                               * NOT CONVERTED *
     HSORTR      5A          X  20                                   * NOT CONVERTED *
     END    1    5                                                   * NOT CONVERTED *  *28*
     FDC    1   20                                                   * NOT CONVERTED *


// LOAD $COPY,R1                                                     * NOT CONVERTED *  *46*
// FILE NAME-COPYIN,UNIT-D1,PACK-D1D1D1,LABEL-FILE3                  * NOT CONVERTED *
// RUN                                                               * NOT CONVERTED *
// COPYFILE OUTPUT-PRINT                                             * NOT CONVERTED *
// SELECT KEY,FROM-'00012'                                           * NOT CONVERTED *


// LOAD SAMPL3,R1                                                    * NOT CONVERTED *  *03*
// FILE NAME-FILE1,UNIT-D1,PACK-D1D1D1,LABEL-FILE3                   * NOT CONVERTED *
// RUN                                                               * NOT CONVERTED *


* * * COPY   EXECUTION * * *
// ASSGN SYS003,FBA,VOL=VOL002,SHR
// DLBL INFILE,'KSDS.FILE2',60,VSAM
```

```
// EXTENT SYS003,VOL002
// EXEC IDCAMS,SIZE=AUTO
   PRINT        INFILE(INFILE) DUMP-
                FKEY(00012)
/*
```

Conversion Aid System/3 to DOS/VSE PDOM Printed in Denmark SB11-5351-0

IBM World Trade
Americas/Far East Corporation
Town of Mount Pleasant
Route 9, North Tarrytown
New York, 10591, USA

IBM Eurocoordination
Tour Franklin
Cedex 11
92081 Paris, La Défense
France

A-COMP ApS 01-54 20 40

READER'S COMMENT FORM

Conversion Aid System/3
to DOS/VSE PDOM

SB11-5351-0

We would appreciate your comments. Please give specific page and line references where appropriate. Your comments will be carefully reviewed by the persons responsible for writing and publishing this material.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

If you wish a reply, please be sure to include your name and address.

COMMENTS

THANK YOU FOR YOUR COOPERATION
PLEASE FOLD ON TWO LINES, STAPLE AND MAIL

YOUR COMMENTS, PLEASE

Your comments on the other side of this form will help us improve future editions of this publication. Each reply will be carefully reviewed by the persons and department responsible for writing and publishing this material.

Please note that requests for copies of publications and for assistance in utilizing your IBM system should be directed to your IBM representative or the IBM branch office serving your locality.

IBM Netherlands
International Field Program Center
P. O. Box 24
1420 AA UITHOORN
Netherlands

Conversion Aid System/3 to DOS/VSE PDOM Printed in Denmark SB11-5351-0

IBM World Trade
Americas/Far East Corporation
Town of Mount Pleasant
Route 9, North Tarrytown
New York, 10591, USA

IBM Eurocoordination
Tour Franklin
Cedex 11
92081 Paris, La Défense
France