SC24-5192-0
File No. S370-30

Program Product

Using the VSE/VSAM Space Management for SAM Feature

Program Number 5746-AM2

Release 2

Preface

The primary purpose of this manual is to provide the information necessary for placing SAM files into VSAM data space when the *VSE/VSAM Space Management for SAM Feature* is installed. The intended audience of this manual is VSAM users who also have SAM files that they would like to have processed and managed by VSAM's facility.

You are presumed to have a knowledge of VSAM job control and Access Method Services commands. A knowledge of SAM is also assumed.

This publication contains:

- An overview of the facilities and functions available for SAM files when the feature is installed.
- An explanation of the functions that become available at the different levels of conversion when changing SAM files to VSAM files.
- A detailed explanation of the commands used for creating, deleting, updating, and extending SAM files in VSAM data space.

This manual also contains an appendix which contains illustrations and accompanying explanations showing the structure of control intervals and control areas. There is also a glossary of terms and definitions as they apply to this particular publication.

Prerequisite Publications:

The reader should be familiar with pertinent information presented in the following publications:

- *VSE/VSAM General Information,* GC24-5143
- *Using VSE/VSAM Commands and Macros,* SC24-5144
- *VSE/Advanced Functions Macro Reference,* SC24-5211

Related Publications:

- *VSE/VSAM Messages and Codes,* SC24-5146
- *VSE/VSAM Programmer's Reference,* SC24-5145

These and other VSE/Advanced Functions publications are described in the *IBM System/370 and 4300 Processors Bibliography,* GC20-0001. Terminology is defined in the *Data Processing Glossary,* GC20-1699.

The VSE/VSAM Space Management for SAM Feature provides the basis for DASD space management of SAM files. Use of this feature with language compilers and utility programs may require you to refer to the publications associated with these programs.

First Edition (December 1979)

This edition, SC24-5192-0, applies to the IBM VSE/VSAM Space Management Feature (Program Number 5746-AM2), and to all subsequent releases until otherwise indicated in new editions or Technical Newsletters. Changes are continually made to the information herein. Before using this publication in connection with the operation of IBM systems, consult the latest edition of *IBM System/370 and 4300 Processors Bibliography,* GC20-0001, for the editions that are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

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 Programming Publications, Dept. G60, P.O. Box 6, Endicott, New York, U.S.A. 13760. 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.

© Copyright International Business Machines Corporation 1979

Contents

Introduction 5
Using VSAM With SAM 5
Ease of Use 6
Dynamic Allocation 6
Simplified Job Control 6
Default Modeling 6
Implicit File Definition 6
DASD Independence 7
Access Method Services Commands 7
Security and Integrity of Data 7
Security and Password Protection 7
Integrity 7
Automatic Close 8
DASD Sharing of SAM Files 8
Data Recovery 8
Additional Functions Available for Managed-SAM Access 8
Work Files 9
Disposition 9
Extending Existing SAM Files 10

Moving Data and Programs from SAM to VSAM Control 11
Levels of Migration 11

How To Use the VSE/VSAM Space Management for SAM Feature 13
Setting Up a Quantity of Space 13
Defining a SAM ESDS 13
Explicit Define via the DEFINE CLUSTER Command 14
Parameter Explanations 14
Additional Considerations 16
Single Extent Primary Allocation 16
Partition/Processor Independence Specification 16
Implicit Define Cluster 16
Assumptions Made by VSAM 17
Information Obtained (by VSAM) from the DTF 17
Information Obtained (by VSAM) from the Job Control Statements 18
Resetting and Reusing a Previously-Defined File 19
Using a SAM ESDS 19
Define and Managed-SAM Access Considerations 19
Differences between Managed-SAM Access and (Unmanaged) SAM Access 20
Define Considerations 20
Access Considerations 21
Restrictions on Using SAM ESDSs 22
VSAM Access Considerations 23
Access Method Services for a SAM ESDS 24
Deleting a SAM ESDS 27
Explicit Delete - DELETE Access Method Services Command 27
Implicit Delete 27

Appendix A: Sample Programs and Job Streams 29

Example 1: Load a SAM ESDS via Managed-SAM Access 29
Example 2: Implicit Define of a SAM ESDS 30
Example 3: Define a Default Model SAM ESDS 31
Example 4: Define a Dynamic SAM ESDS and Access 32

Appendix B: Differences between (VSAM) ESDS and SAM ESDS Format 34

Glossary 38

Index 41

Figures

Figure 1. Ways to use SAM with VSAM 6
Figure 2. Managed-SAM Access OPEN Disposition 10
Figure 3. Managed-SAM Access CLOSE Disposition 10
Figure 4. Migration from SAM to VSAM Control 12
Figure 5. Valid Combinations of Access Methods and File Types 12
Figure 6. Example of CA Format Using a VSAM Entry-Sequenced File 34
Figure 7. Example of Non-CA Format Using a SAM ESDS 35
Figure 8. Comparison of a VSAM Logical Record to a SAM Logical Block 36

Using VSAM with SAM

Prior to VSE/VSAM Release 2, DOS/VS and DOS/VSE users had to be concerned with planning, organizing, and maintaining space for files accessed by SAM. You had to consider the specific location of files, DASD characteristics, block sizes and blocking factors. This involved considerable manual control which, in a multi-programming environment, presented a complex problem. This burden is removed by a DASD space management technique called *VSAM Space Management for SAM*.

The VSE/VSAM Space Management for SAM Feature is an enhancement of VSAM's space management. It allows you to define and process your SAM files within VSAM data space, using a DTF and the SAM imperative macros (for example: OPEN, GET, PUT, etc.). This type of file is called a SAM ESDS. A SAM ESDS can also be accessed by VSAM(ACB) access.

Note: A SAM ESDS is not identical to a (VSAM) ESDS (see Appendix A). A new parameter (RECORDFORMAT) is specified on the Access Method Services DEFINE CLUSTER command for a NONINDEXED (ESDS) file to indicate a SAM ESDS and provide SAM record format information.

You can convert your SAM files to VSAM files. There are three steps involved in the change, with the option of being able to stop at any level. They are:

1. Move your unmanaged SAM files into VSAM data space (they become SAM ESDSs and can be accessed by managed-SAM).
2. Change your programs from managed-SAM access to VSAM access of a SAM ESDS.
3. Convert your SAM ESDS data files to VSAM entry-sequenced files (VSAM access only).

Depending on the level of change, certain functions and capabilities are available for processing the files. Files that have been converted to VSAM can take full advantage of VSAM's processing capabilities. Refer to the section entitled "Moving Data and Programs from SAM to VSAM Control".

During VSAM access of a SAM ESDS, SAM records are processed according to the RECORDFORMAT specified at DEFINE time. To the VSAM program it appears as though they are in VSAM entry-sequenced file format. An existing VSAM program that sequentially processes a VSAM entry-sequenced file can access a SAM ESDS.

This feature allows you to request quantities of DASD space rather than absolute locations. You indicate your intention to use a SAM file in VSAM data space by opening a DTF that specifies a filename described in a VSAM DLBL statement. This tells open that the file to be accessed is a SAM ESDS. In this case, managed-SAM open retrieves file information from the VSAM catalog rather than the VTOC.

Data is written in a format similar to a VSAM ESDS. CI (control interval) format is used and the CI is the basic unit of information that is transmitted to or from a direct access device. This format allows DASD independence (for example, maximum DTF BLKSIZE is not limited to DASD track size but only to CI size minus 7) and allows VSAM access (via ACB) to SAM files in VSAM data space. You need not specify absolute extent limits for the file because VSAM determines the location of the file. Figure 1 illustrates three ways to take advantage of the SAM/VSAM combination: when using work files, data files, or when moving data and programs from SAM to VSAM control.

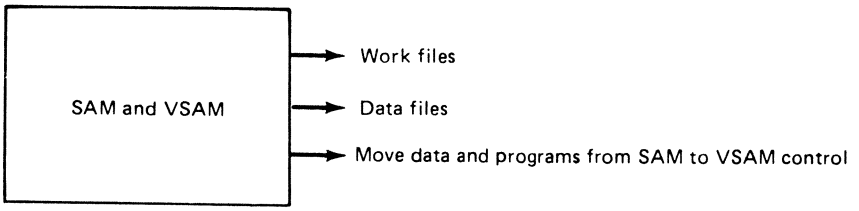

Figure 1. Ways to use SAM with VSAM

This support includes all DASD devices (CKD and FBA) that are supported by VSAM. It includes DTFSD (data and work files) and DTFPH for DASD with MOUNTED=SINGLE.

Ease of Use

With this feature, you can take advantage of many of the DASD space management (and other) functions available in VSAM, along with Access Method Services Commands.

Dynamic Allocation: With VSAM managing your space, you can take advantage of VSAM's dynamic allocation. The allocation of file space is simpler because you do not have to specify extent limits. You need only request a quantity of space. This space is allocated when it is needed. If more space is subsequently needed, a secondary quantity is allocated.

VSAM's dynamic file capability allows you to define a file in the VSAM catalog without allocating space for it. Space is allocated at open and deleted at close under control of the DLBL DISP parameter. This dynamic file capability applies to SAM ESDSs.

Simplified Job Control: Previously job control statements were necessary to assign logical units to physical devices and notify the system that a specific volume was mounted on a logical unit. It was the responsibility of the programmer to supply this information to the system by way of the ASSGN and EXTENT job control statements. With VSE/VSAM support, this task is significantly simplified by eliminating the need for ASSGN and EXTENT statements in most cases. (Refer to "Chapter 4: Job Control" in *VSE/VSAM Programmer's Reference* for restrictions involving the use of job control statements.) This improvement is available to the SAM user through the VSE/VSAM Space Management for SAM Feature. The information required by the system to check the location and characteristics of files is stored in the VSAM catalog. The need for DLBL statements is also removed for many of the Access Method Services commands. Because VSAM user catalogs are programmer logical units, they too are eligible for automatic assignment. Operator communications are also simplified because the operator may mount a requested volume on an available drive without the need to assign the drive.

Default Modeling: Default modeling allows you to select your own parameter defaults in place of the usual system defaults during explicit define. The ability to specify default parameters for the Access Method Services DEFINE CLUSTER command via default modeling is available for SAM ESDSs as well as for VSAM files.

Implicit File Definition: SAM ESDSs do not have to be explicitly defined (Access Method Services DEFINE Command) prior to the time they are opened. An implicit define of a reusable SAM ESDS occurs during managed-SAM open if the file has not yet been explicitly defined. Generally, when a file is implicitly defined, it may be

implicitly deleted during managed-SAM close depending on the disposition parameters specified on the DTF or DLBL statement.

DASD Independence: You do not have to be concerned with different track and cylinder sizes for various types of devices (the DTF DEVICE and DEVADDR parameters are ignored by managed-SAM open so that the file may reside on any DASD device type). Allocation sizes may be requested in terms of number of records and average record length rather than tracks and cylinders, which are device dependent. This may be specified on the DEFINE CLUSTER command for explicitly defined files or on the DLBL RECORDS and RECSIZE parameters for implicitly defined files. For implicitly defined files, a default secondary allocation size of twenty percent of the primary allocation size (rounded up) is assumed if none is specified.

The CI is the basic unit of information that VSAM transmits to or from a direct access device. Because the CI size has no relation to the track and cylinder size of a particular device, this makes the processing of files DASD independent.

Access Method Services Commands: You no longer need to use different utility programs to manipulate files. With the VSE/VSAM Space Management for SAM Feature, the Access Method Services commands can be used to print, copy, alter, delete, and move files from one system to another. Additional information appears in the section "Access Method Services for a SAM ESDS".

Security and Integrity of Data

VSAM assures the security and integrity of data through a combination of VSAM facilities. These include the share options support for SAM ESDSs to control access to data files, password protection to prevent unauthorized access to both the catalog and data files, and the automatic close facility for data files that were not closed prior to the end of job step.

Security and Password Protection: VSAM password protection is provided for explicitly defined managed-SAM files. This option enables you to protect your files from unauthorized use. Files can be protected at various levels of security with each level requiring a separate password. The password for accessing your managed-SAM files is provided by operator response (since there is no way to specify a password on a DTF). Password information is kept in the VSAM catalog. If you do not specify the password protection option, no password is required to access your file.

Integrity: Managed-SAM access has increased integrity over native SAM access in the following ways:

1. VSAM controls the space that is available for allocation. Files cannot overlap because extent limits are never specified.
2. In general, when a SAM ESDS is opened for output or work it is reset to empty. (Refer to Figure 2 for specific cases when a SAM ESDS is not set to empty.) However, VSAM will not reset a file when it is still in use by another user for either input or output, regardless of the sharing options specified for the file and regardless of expiration date. Therefore, it is not possible to delete a file while another user has the same file open for access as may occur in native SAM.
3. In addition, after a SAM ESDS is open for output or extension, SAM users who wish to open a file for input (with or without update) are denied access to the file. This is to keep the input user from reading past the end of the file since the SEOF (Software End Of File) does not normally exist until the output file is closed (SAM ESDSs run in SPEED mode during load and extend). This occurs

regardless of the sharing options specified for the file. Conversely, when a SAM ESDS is open for input, an output or extend user will be denied access to the file for the same reason.

4. Except for the conditions specified in 2 and 3, VSAM controls access the file according to the sharing options specified or defaulted. The default is SHAREOPTIONS (1) which allows only one output user or many input users to have the file open at a time.

A specification of SHAREOPTIONS (4) on an explicit define of a SAM ESDS is treated as SHAREOPTIONS (2), as it is for a VSAM entry-sequenced file. Track hold and block hold are controlled by the HOLD=YES specification on the DTF.

Automatic Close: If your SAM ESDS or VSAM file has not been closed prior to the end of job step, or your program fails to execute properly, VSAM's automatic CLOSE facility attempts to close your file for you. This ensures that the access method has written the last logical block of data and the SEOF. If the close should fail, a warning message is issued the next time an attempt is made to reopen the file. The Access Method Services VERIFY command can be used to reset the "file not successfully closed" condition.

DASD Sharing of SAM Files: Currently, VSAM files can be shared across partitions within a system in one of four ways. The level of shared access allowed for a file is an attribute of the file itself. This is determined by the SHAREOPTIONS value chosen when the file is defined. DASD Sharing supports a subset of SHAREOPTIONS for sharing SAM ESDS as well as VSAM files across VSE systems. Depending on the SHAREOPTIONS value chosen, various levels of read and write integrity are ensured by controlling access at the file level between systems. These capabilities apply to a SAM ESDS just as they do to a VSAM entry-sequenced file for both SAM and VSAM access.

Data Recovery

Data recovery is supported for a SAM ESDS through various Access Method Services commands. You can use the EXPORT(RA) and IMPORT(RA) commands in the same way as for a VSAM file; that is, to create a portable copy of a SAM ESDS via the EXPORT command and to introduce the copy of the file into the system using the IMPORT command as a basis for reconstruction if the original file becomes inaccessible.

The LISTCRA command can be used to list catalog recovery areas or compare catalog recovery areas with catalogs. Catalog recovery is also supported via the RESETCAT, REPRO, and EXPORTRA Access Method Services commands.

Additional Functions Available for Managed-SAM Access

In addition to the specific areas of improvement mentioned, these additional functions are available to facilitate processing of SAM ESDSs:

- SAM ESDSs have the capability to support multiple extents and multiple volumes unless:
  - during definition there was no secondary allocation size specified or a single volume was specified,

    or

  - the program accessing the file does not support multiple extents (for example, DTFPH with EXCP access).
- The ability to extend a SAM ESDS by way of SAM data file access is requested via the disposition parameter (DISP=OLD) in the DLBL statement.

Work Files

Work files may need varying amounts of space for different jobs. For some jobs a small quantity of space is all that is needed. At other times a great deal of space is needed. Until now you had to take this into account to ensure that enough space was available for jobs requiring large quantities of space.

The automatic space management capability provided by VSAM enables you to use space more efficiently. No longer do you have to keep large amounts of space tied up. The space needed for work files can be smaller than before; you can think in terms of the average size of space needed rather than the maximum size needed. This is because VSAM provides dynamic secondary allocation. You can make your primary allocation nearer to the average size of space needed; if more is needed VSAM gets the necessary space by using the secondary allocation.

In addition to dynamic secondary allocation, VSAM provides dynamic primary allocation. This capability allows you to define a file that needs no space until it is opened (dynamic file). When the file is opened, the needed space is provided by VSAM. The options available at OPEN and the disposition of the files at CLOSE depend on what you code in the DLBL statement DISP parameter or what the DTF has specified (for example: DELETFL=NO).

Partition/processor independence, through the implicit define or explicit define with the dynamic data set capability, eliminates the task of assigning different work files to different DASD locations for each partition, and having to specify those locations in your job control statements. The file-ID is chosen according to the partition that the job is running in, and space is assigned as needed. Work-space can also be shared between processors with the capability to run the same job in any partition of a number of different processors without conflict in the catalog. When the file is closed it may be deleted or deallocated. The space the file occupied is reclaimed and made available for use by other files.

All the functions described in the section "Using VSAM with SAM" apply to work files, but you may find some (such as password protection and data recovery) inappropriate.

Disposition

The disposition of a reusable file (REUSE) can now be controlled through the DLBL DISP parameter specification. A file can be allocated, reset, or implicitly defined at open according to these specifications. Whatever you specify in the DLBL DISP parameter overrides whatever was specified on the DTF. Pertinent information from the disposition parameters and the DTF is saved for closing of the file. At that time, the file is kept, reset, deallocated, or deleted according to the disposition that is specified on the DTF and DLBL statement. When you do not specify the DISP parameter, a default is chosen according to the type of file being opened or closed. The default disposition is the same as would occur for unmanaged SAM files. For example, the default disposition for:

- DTFSD OUTPUT data file is DISP=(NEW,KEEP)
- DTFSD INPUT data file is DISP=(OLD,KEEP)
- DTFSD work file is DISP=(NEW,DELETE).
  If DELETFL=NO, then DISP=(NEW,KEEP).

Refer to Figures 2 and 3 for disposition parameter specifications and their results. Disposition processing for VSAM (ACB) access of a SAM ESDS is the same as for a VSAM entry-sequenced file. *VSE/VSAM Programmer's Reference* contains additional information about the DLBL statement and DISP parameter processing.

Extending Existing SAM Files

With VSAM support you can also extend existing SAM ESDSs through the use of the DLBL DISP parameter. For example (referring to Figure 2), to extend a SAM ESDS during output processing using SAM access, DISP=OLD would position you to the end of the file to allow for extension. Space for extension of a file is allocated (if necessary) according to the secondary space allocation specified at definition time. This support is not provided for work file access. SAM ESDS files are always extended in SPEED mode. (Refer to Example 4. in Appendix B.)

Files with REUSE Attribute				
OPEN (DTF)	File Status	DISP Unspecified (Default Value in Parentheses)	DISP on DLBL[2] Statement	
			NEW	OLD
Output	Unallocated SAM ESDS[1]	Allocate space for file. (DISP=NEW)	Allocate space for file.	Allocate space for file.
	Allocated for SAM ESDS[1]	Reset file to its primary allocation, release secondary extents, indicate file as empty. Position to beginning of file. (DISP=NEW)	Reset file to its primary allocation, release secondary extents, indicate file as empty. Position to beginning of file.	File is not reset to empty. Position to end of file for extension.
	Undefined	Implicitly define a SAM ESDS. (DISP=NEW)	Implicitly define a SAM ESDS.	Implicitly define a SAM ESDS.
Input	Allocated for SAM ESDS	File is not reset to empty. Position to beginning for input. (DISP=OLD)	Invalid - file is not reset. Open fails	File is not reset to empty. Position to beginning for input.
Work	Unallocated SAM ESDS[1]	Allocate space for file. (DISP=NEW)	Allocate space for file.	Allocate space for file.
	Allocated for SAM ESDS[1,3]	Reset file to its primary allocation, release secondary extents, indicate file is empty. Position to beginning of file. (DISP=NEW)	Reset file to its primary allocation, release secondary extents, indicate file is empty. Position to beginning of file.	File is not reset to empty. Position to beginning of file (file may be read).
	Undefined	Implicitly define a SAM ESDS. (DISP=NEW)	Implicitly define a SAM ESDS.	Implicitly define a SAM ESDS.

1 If the file characteristics do not match those specified in the DTF, the open fails if the file cannot be implicitly deleted. In particular, the maximum logical block that may be written (DTF BLKSIZE) must not be greater than the maximum allowed in the file (maximum RECORDSIZE). Also, if DTFSD is used, the file must be in CI format.

2 Do not specify the DISP parameter for IJSYSLN (SYSLNK file).

3 DISP=NEW prevents access to any data existing prior to open.

Figure 2. Managed-SAM Access OPEN Disposition

File with REUSE Attribute					
CLOSE (DTF)	DISP Unspecified	DISP on DLBL[2] Statement			
		KEEP	DELETE	DATE	
				Expired	Unexpired
File was explicitly defined Dynamic (NOALLOCATION)	Keep[1]	Keep	Deallocate	Deallocate	Keep
Reusable	Keep[1]	Keep	Reset	Reset	Keep
File was implicitly defined	Keep[1]	Keep	Delete	Delete	Keep

1 DISP is Delete if TYPEFLE=WORK and DELETFL is unspecified.

2 Do not specify the DISP parameter for IJSYSLN (SYSLNK file).

If you request DELETE at close, VSAM will delete the data by way of deallocation, reset, or implicit delete. However, in order to avoid sharing problems, if any other DTF or ACB for the same file is open at the same time, no DELETE occurs, the operator is notified by a message with a warning return code, and the close will continue processing. With DELETE specified at close, the contents of the file are lost. The next open for OUTPUT or WORK will write new data. If the file has been deallocated or reset, an open for INPUT will be successful, but the user will be sent to the EOFADDR routine on the first GET because the file is empty.

Figure 3. Managed-SAM Access CLOSE Disposition

Moving Data and Programs from SAM to VSAM Control

This new support can be considered a bridge to allow the migration of SAM files and SAM programs to VSAM files and VSAM programs.

Levels of Migration

Moving from SAM to VSAM control consists of up to three steps which are illustrated by Figure 4. They are:

1. Move unmanaged-SAM data (files) into SAM ESDSs. This allows managed-SAM access.
2. Change programs to VSAM access from managed-SAM access (SAM ESDSs are accessible by VSAM).
3. Convert data (files) from SAM ESDSs to VSAM entry-sequenced files (files accessible by VSAM only).

Step (1): When you take the first migration step (move SAM files to SAM ESDSs), you get all the new functions previously described in this manual.

Step (2): When you take the second migration step (change managed-SAM access programs to VSAM access programs), you get these additional access functions provided by VSAM access:

- VSAM provides a single ACB/RPL format and a single set of request macros for all file types. You can generate the ACB or the RPL by specifying the GENCB macro.
- The file can be accessed in a direct manner via access by RBAs.
- The file can be processed in a skip sequential manner or sequentially backwards.
- Access statistics are maintained, displayable via LISTCAT.
- Multiple (CI) buffers may be used in support of VSAM's read-ahead capability.
- A password may be specified on the ACB so that the operator is not involved with passwords.
- The job is not canceled due to logical or physical errors as in SAM. Rather, a return code and error code are passed back to the user to allow diagnosis of the failure within the user's program.
- Multiple strings and chained RPL support are provided by VSAM.

Step (3): When you take the final migration step (convert SAM ESDSs to VSAM entry-sequenced files), these additional capabilities become available to you:

- You are able to build alternate indexes or paths which allow you other ways of gaining access to your files, thereby eliminating the need to keep multiple copies of the same information sorted differently for different applications.
- RECOVERY can be specified in the DEFINE CLUSTER statement. This parameter will help ensure data integrity by preformatting each control area before records are loaded into it. In case of load or extend failure, the Access Method Services VERIFY command can be used to recover data written, and your program may resume writing data from the last correctly-written data record.
- Records can be spanned (SPANNED) records, eliminating the need for very large control intervals.

- Generally, a VSAM entry-sequenced file is portable to OS/VS and can be accessed via OS VSAM. Refer to *VSE/VSAM General Information Manual* for specific cases when files are not portable.

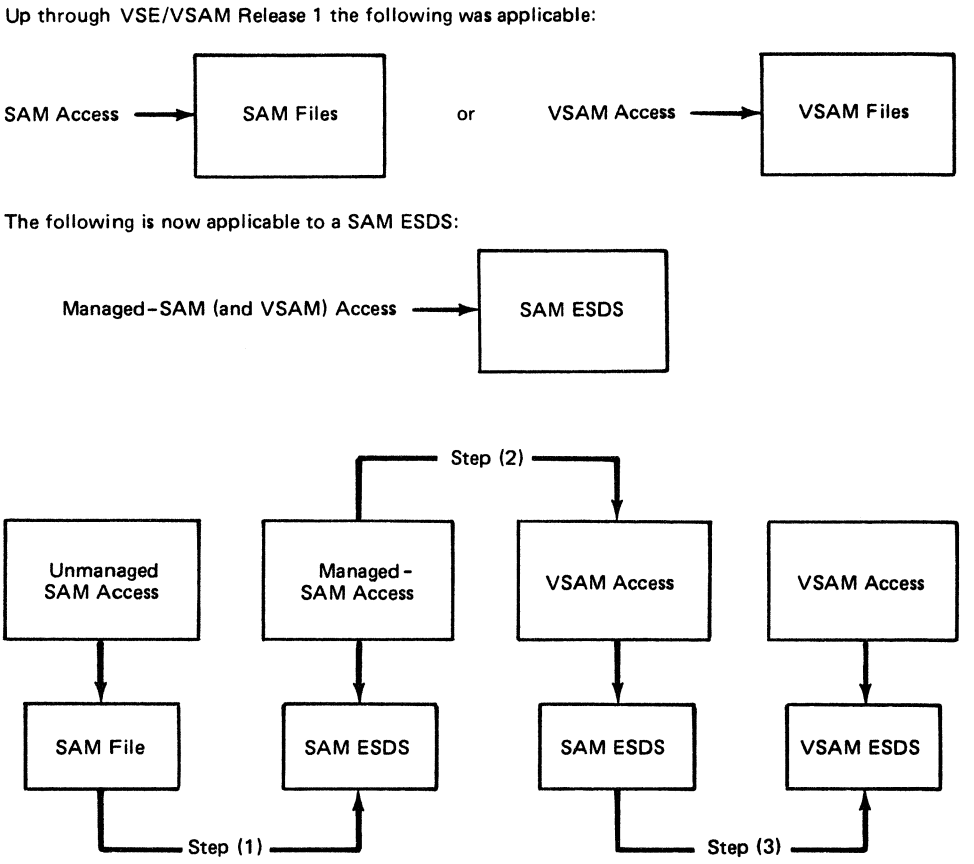

Figure 4. Migration from SAM to VSAM Control

Figure 5. illustrates the valid combinations of access methods and file types that can be used with the VSE/VSAM Space Management for SAM Feature installed.

FILE TYPE \ ACCESS MODE	VSAM ACCESS	Managed-SAM ACCESS
VSAM entry-sequenced file	VALID	INVALID
SAM ESDS	VALID (NEW)	VALID (NEW)

Figure 5. Valid Combinations of Access Methods and File Types

How To Use the VSE/VSAM Space Management for SAM Feature

To use the VSE/VSAM Space Management for SAM Feature you must perform the operations listed below. This chapter explains what each step is and what you must do for it.

1. Set up a quantity of space
   - Define MASTERCATALOG command
   - Define USERCATALOG command
   - Define SPACE command.
2. Define a SAM ESDS
   - Explicit define (via Access Method Services)
   - Implicit define (via managed-SAM open).
3. Access the file
   - Using DTF and SAM imperative macros with VSAM DLBL
   - Using ACB and VSAM imperative macros with VSAM DLBL
   - Using Access Method Services commands with VSAM DLBL.
4. Delete the file (optional)
   - Explicit delete (via Access Method Services)
   - Implicit delete (via managed-SAM close).

Setting up a Quantity of Space

Space for a SAM ESDS may be suballocated by VSAM out of data space that was previously defined for VSAM files. You need not assign separate space for SAM ESDSs. The size and boundaries of the suballocated space are communicated to the managed-SAM access routines at open and secondary allocation time. You define this space (ideally entire volumes) in the usual way by using one or more of the following Access Method Services commands:

- DEFINE MASTERCATALOG
- DEFINE USERCATALOG
- DEFINE SPACE.

See *Using VSE/VSAM Commands and Macros* for information about these commands.

Note: The definition of data space and catalogs may be handled by installation management or the system programmer. If this is the case, the application programmer need not be concerned with this step.

If the catalog is password protected, implicit define will request the update or higher level password of the catalog and implicit delete will request the master password of the catalog.

Defining a SAM ESDS

After defining sufficient VSAM data space, you define your SAM ESDS file. There are two ways of doing this:

- Explicit define, that is, by using the Access Method Services DEFINE CLUSTER command.
- Implicit define, that is, by providing the required file information in the job control statements so that the file can be defined at managed-SAM open.

There is also a third possibility which cannot actually be classified as defining a file. That is, you can take a file that has already been defined (either explicitly or implicitly) and used, extend it, or reset it to empty and reuse it.

Explicit Define via the DEFINE CLUSTER Command

The following is not a complete list of the parameters that are eligible to be specified for a SAM ESDS, but these particular parameters must be clarified for a SAM ESDS.

You indicate that a SAM ESDS is to be defined by specifying the following parameters in the DEFINE CLUSTER command:

- NAME(entryname) - cluster level
  (Required parameter)
- NAME(entryname) - data component level
  (Optional parameter unless you wish to request single extent primary allocation, in which case it is required)
- NONINDEXED
  (Required parameter)
- RECORDFORMAT(format[(logicalrecordsize)])
  (Required parameter)
- RECORDSIZE(average maximum)
  (Required parameter if RECORDFORMAT is in fixed format; for example, FIXUNB or FIXBLK. Optional for V, VB, or U format.)
- RECORDS(primary[secondary])
  or
  TRACKS(primary[secondary])
  or
  CYLINDERS(primary[secondary])
  or
  BLOCKS(primary[secondary])
  (One of these parameters is required unless a default model exists for a SAM ESDS.)
- VOLUMES(volser[volser...])
  (Required parameter unless a default model exists for a SAM ESDS.)

Parameter Explanations:

NAME(entryname): Specifies the file-ID of the SAM ESDS. For a single extent primary allocation you must specify both the cluster name and the data component name. (See the section entitled "Single Extent Primary Allocation".) Otherwise, the data component name is optional and if specified, can be any name. Also, you can specify that a file be partition independent, or both partition independent and processor independent (see the section entitled "Partition/Processor Independence Specification".) Specifying file names at both levels (cluster and data) gives you the capability to access data under two different file-IDs.

NONINDEXED: Specifies that the file being defined is an entry-sequenced file.

RECORDFORMAT(format[(logicalrecordsize[1])]): Establishes a NONINDEXED file as a SAM ESDS. This parameter is required to explicitly define a SAM ESDS; you can specify it either at the cluster or data component level.

For *format*, substitute one of the following values:

Format	Abbreviation	Meaning	Type of Access
FIXUNB	F	Fixed, unblocked	Managed-SAM/VSAM
FIXBLK	FB	Fixed, blocked	Managed-SAM/VSAM
VARUNB	V	Variable, unblocked	Managed-SAM/VSAM
VARBLK	VB	Variable, blocked	Managed-SAM/VSAM
UNDEF	U	Undefined	Managed-SAM/VSAM
NOCIFORMAT	NCIF	See below[2]	See below[2]

FIXUNB, FIXBLK(logicalrecordsize), VARUNB, VARBLK, and UNDEF indicate that data records are stored in CI format and therefore are accessible and managed by VSAM.

[1] Logicalrecordsize indicates the length of the SAM logical record. This value must always be specified when using FIXBLK format.

[2] NOCIFORMAT indicates that data is not stored in control interval (VSAM) format; therefore DTFPH with physical I/O (EXCP), and not managed-SAM access or VSAM access, must be used to access the data records. This method of access should only be used for local (work) files. Other SAM programs will not be able to read or write to the file (except for other programs that have been written specifically for NOCIFORMAT access; for example, EXCP). You cannot specify NOCIFORMAT together with any of the following parameters: CONTROLINTERVALSIZE, ERASE, BUFFERSPACE, EXCEPTIONEXIT, or WRITECHECK.

RECORDSIZE(average maximum) and RECORDS(primary[secondary]): When you specify the RECORDFORMAT parameter with the FIXUNB or FIXBLK subparameter, you must specify the *maximum* SAM logical block size in the RECORDSIZE(maximum) parameter.

Note: This parameter specifies the largest SAM logical block size that may be used. If a DTF is opened for OUTPUT or WORK and specifies a BLKSIZE larger than the maximum SAM logical block size allowable in the file, the open fails and the job is canceled. You must be careful to specify the maximum RECORDSIZE that a system program or program product will use during explicit define of the file. If multiple system programs or program products are to use the same (work) file, the maximum RECORDSIZE should be equal to the largest record that any of the programs will use.

If the RECORDFORMAT parameter is specified as VARUNB, VARBLK, or UNDEF and the RECORDSIZE parameter is omitted, the RECORDSIZE defaults to 4089 for the average and 4089 for the maximum (that is, RECORDSIZE(4089 4089)). (Note that RECORDSIZE(maximum) is used in calculating the control interval size and therefore has no meaning when NOCIFORMAT is specified.)

Whether or not you specify the NOCIFORMAT subparameter, you can use the RECORDSIZE(average) and RECORDS parameters for the suballocation of space. When using the RECORDSIZE and RECORDS parameters together, they must be consistent in units of reference (either both refer to SAM logical records or both refer to SAM logical blocks). Note that both the average and maximum recordsize must be specified on the RECORDSIZE parameter when one is specified.

For V, VB, or U records, the RECORDSIZE parameter is optional. For F or FB records, the RECORDSIZE parameter is required. For FB records, the RECORDSIZE must be a multiple of the SAM logical recordsize specified in the RECORDFORMAT parameter. For V or VB records, the maximum RECORDSIZE parameter must include room for the control information for variable length records (the record length field is four bytes and the block length field is four bytes) because the control information is part of the SAM logical block.

TRACKS|CYLINDERS|BLOCKS(primary[secondary]): The rules involved in the use of these parameters are the same for a SAM ESDS file as for a VSAM file. See *Using VSE/VSAM Commands and Macros* for information concerning the TRACKS, CYLINDERS, and BLOCKS parameters.

VOLUMES(volser[volser...]) - Optional: Specifies the volume(s) to contain the SAM ESDS. Each volume that you specify must be owned by the catalog that is to own the SAM ESDS. If not specified during define of a SAM ESDS, VSAM picks a set of volumes for you if you have a default model (DEFAULT.MODEL.ESDS.SAM) defined. See *VSE/VSAM Programmer's Reference* for information on defining default models and *Using VSE/VSAM Commands and Macros* for information on the VOLUMES parameter.

Additional Considerations

- The RECORDFORMAT attributes can be modeled via the MODEL parameter.
- You should specify REUSE when a SAM ESDS is used mainly for work files. You should additionally specify NOALLOCATION in the DEFINE CLUSTER command to provide the dynamic file capability to work files.
- Do not specify RECOVERY. (VSAM defaults to SPEED for a SAM ESDS.) You cannot build an alternate index or define a path over a SAM ESDS.

Note: For work files a zero retention period is the default and is normally appropriate to avoid operator communications during a subsequent open if the file was not deleted at close.

Single Extent Primary Allocation

NAME(DOS.WORKFILE.SYSentryname) - data component level: Some programs that access data via DTFPH with EXCP may require that DASD space for the file be allocated as a single extent. You can specify that you want the primary space allocated as a single extent by specifying the data component name as above. (Normally, VSAM may obtain an allocation in as many as five extents.) The cluster name is still chosen in the same manner as before but DOS.WORKFILE.SYS *must* prefix the *data* component name in order to ensure that space is allocated within a single extent.

VSAM will deny the allocation request if it cannot obtain the primary allocation in a single extent.

Partition/Processor Independence Specification

NAME(%entryname) - Partition independent file-ID:
NAME(%%entryname) - Partition and processor independent file-ID:

You specify a partition-unique file-ID by using the prefix "%" on the cluster name in the DEFINE CLUSTER command. (The file-ID is limited to twenty-seven characters in this case.)

To specify both a partition-unique and processor-unique file-ID together with a single extent primary allocation, the cluster name must be prefixed with "%%" (the file-ID is limited to twenty-seven characters in this case) and the data component name must be prefixed with "%%DOS.WORKFILE.SYS" (the file-ID is limited to an additional eleven characters in this case).

If your system also has Interactive Computing and Control Facility (ICCF) installed, you are allowed only one partition-independent file for each ICCF real-partition. ICCF pseudo-partitions do not have unique partition IDs, so there can be only one partition-independent file per partition.

Implicit Define Cluster

A SAM ESDS can be defined implicitly through managed-SAM open when TYPEFLE=OUTPUT or WORK is specified on the DTF. An implicit define cluster occurs as a result of the following two conditions:

- The SAM ESDS (to be opened via the DTF and written to) is currently unde-

fined in the VSAM catalog or the characteristics of the file were not compatible with the DTF (see note on Figure 2) and the file has been implicitly deleted by open.

- Enough information has been provided for the implicit define to occur. VSAM gathers the necessary information from three sources:
  - — it makes several assumptions about the file (see below)
  - — it extracts information from the DTF specifications
  - — it extracts information from the job control statements.

Assumptions Made by VSAM

VSAM always makes the following assumptions for an implicit define:

Parameter	Assumption
NONINDEXED	An entry-sequenced file is being defined.
NONSPANNED	The maximum length of a SAM logical block must not be greater than the control interval size minus 7.
NOWRITECHECK	VSAM access will not check for correct data transfer for records written to the file. (In managed-SAM access, checking for correct data transfer is controlled by the DTF VERIFY=YES or NO parameters. The NOWRITECHECK specification has no meaning for the managed-SAM user. It is used only during VSAM access.)
REUSE	It is possible for a user to reset an already existing file back to empty and reuse it.
SHAREOPTIONS(1 3)	Either any number of users are permitted for input processing, or 1 user is permitted for output processing.
SPEED	Direct access storage is not preformatted.
SUBALLOCATION	VSAM data space for the file was previously defined and a primary allocation is suballocated at define.
UNORDERED	The volumes need not be used in the order specified via the EXTENT job control statements or the default model if EXTENT statements are omitted.
USECLASS(0 P)	The file occupies class-0 data space.

Information Obtained (by VSAM) from the DTF

VSAM extracts information either from the:

- DTFSD
- DTFPH MOUNTED=SINGLE for DASD

to determine:

- — control interval size
- — the length of the maximum VSAM logical record (SAM logical block)
- — the record format of the records in the file
- — the SAM logical record size for FIXBLK.

DTFSD Specifications (for data files)

- Control interval size — derived from the CISIZE=nnnnn parameter. VSAM rounds this value up to a valid CISIZE before defining the file; if zero or no value was specified, VSAM chooses a control interval size. If IOAREA2 is specified and CI size is not specified, VSAM attempts to choose a CI size that insures that at least 2 logical blocks will fit into a CI.
- Maximum record size — derived from the BLKSIZE=nnnn parameter. This value (minus 8 for data OUTPUT DTFs) specifies the file's maximum RECORDSIZE.
- Record format — derived from the RECFORM=xxxxxx parameter. Specifies the RECORDFORMAT of the file. If the RECFORM=FIXBLK, the SAM logical record size is derived from the DTF RECSIZE=nnnnn parameter.

DTFSD Specifications (for work files)

- Control interval size — derived from the CISIZE=nnnnn parameter. VSAM rounds this value up to a valid CISIZE before defining the file; if zero or no value was specified, VSAM chooses a control interval size.
- Maximum record size — derived from the BLKSIZE=nnnn parameter. This value specifies the file's maximum RECORDSIZE.
- Record format — derived from the RECFORM=xxxxxx parameter. Specifies the RECORDFORMAT of the file; FIXUNB and UNDEF are the only valid sub-parameters that you can specify for work files.

DTFPH MOUNTED=SINGLE for DASD

- Control interval size — derived from the CISIZE=nnnnn parameter. If a non-zero value is specified, VSAM rounds this value up to a valid CISIZE before defining the file. Specifying zero is the same as not specifying a CISIZE. In this case, VSAM indicates that the file is non-CI format; it is accessible only by EXCP (not by VSAM or managed-SAM).
- Maximum record size —
  - — if CI format, maximum equals the DTF CISIZE minus 7.
  - — if non-CI format, this parameter does not apply.
- Record format —
  - — if CI format, the RECORDFORMAT is UNDEF.
  - — if non-CI format, the RECORDFORMAT is NOCIFORMAT.

Note: DTFPH (with a CISIZE of zero specified or no CISIZE specified) is the only possible way you can implicitly define a non-CI format file. Also, if a nonzero value is specified for the CISIZE parameter, it must be greater than seven in order to choose a valid maximum record size.

Information Obtained (by VSAM) from the Job Control Statements

Certain operands in the VSAM DLBL and EXTENT job control statements provide information VSAM needs to implicitly define a file. Refer to "Chapter 4: Job Control" in *VSE/VSAM Programmer's Reference* for more information.

The DLBL statement provides the following information for implicit define:

- file-ID — This operand provides the unique name associated with the file. To request single extent allocation through an implicit define, DOS.WORKFILE.SYS must prefix the file-ID. A partition/processor unique file-ID may also be specified. In this case the DLBL file-ID must be specified with a prefix of "%" (partition-unique) or "%%" (partition- and processor-unique) with a limit of twenty-seven characters. For both partition/processor uniqueness and single extent primary allocation, the DLBL file-ID prefix may be specified as "%%DOS.WORKFILE.SYS" (with a limit of eleven additional characters).

  If your system also has Interactive Computing and Control Facility (ICCF) installed, you are allowed only one partition-independent file for each ICCF real-partition. ICCF pseudo-partitions do not have unique partition IDs, so there can be only one partition-independent file per partition.
- date — This operand indicates either the retention period in days or the actual expiration date. If this operand is not present the normal default applies.

Note: For work files, specify a zero (retention period) to avoid operator communications during a subsequent open if the file was not deleted at close.

- CAT=filename — This operand indicates the catalog that owns the file. If this operand is not present the normal default applies.
- RECORDS=(primary[,secondary]) — This operand designates the number of SAM logical records for allocation purposes. This operand must be specified if

the "number of tracks or blocks" operand is omitted from the EXTENT statement. If no secondary amount is specified, twenty percent of the primary allocation is assumed. Zero may be specified for the secondary amount. If RECORDS is specified, RECSIZE=n must also be specified.

- RECSIZE=n — This operand indicates the average SAM logical record size; it must be specified together with the RECORDS=(primary[,secondary]) operand.

Note: You may alternatively specify the average SAM logical *block* size in the RECSIZE parameter, but in this case, you should specify the number of SAM logical *blocks* in the RECORDS parameter.

The EXTENT statement provides the following information for implicit define:

- Volume serial number — This indicates the volume that this file resides on. There must be one EXTENT statement for each volume that the file is eligible to reside on. If EXTENT statements are specified, this parameter is required on each EXTENT statement.
- Number of tracks or blocks (specified on the first EXTENT statement if multiple EXTENT statements are specified) — This indicates the number of tracks (CKD) or blocks (FBA) to be allocated to this file. A secondary allocation size of twenty percent of the primary allocation size (rounded up) is assumed. Whether it is tracks or blocks is determined by the device type of the volume serial number specified. This parameter is ignored on subsequent EXTENT statements, or if the RECORDS/RECSIZE parameters are specified on the DLBL.

Note: The EXTENT statement is not required for implicit define if a default model for a SAM ESDS was previously defined (providing VOLUME information) and RECORDS/RECSIZE are specified on the DLBL statement (providing allocation information).

Resetting and Reusing a Previously-Defined File

You can indicate that a file is to be reset and reused by specifying the NOALLOCATION parameter together with the REUSE parameter in the DEFINE CLUSTER command. This specification indicates that data space is not to be suballocated to the file at DEFINE, rather, it is to be suballocated as needed. This type of file is called a dynamic file.

A file may be implicitly defined at open and implicitly deleted at close. However, performance is not as good as for an explicitly defined dynamic file.

Using a SAM ESDS

Define and Managed-SAM Access Considerations

Managed-SAM access is provided for a SAM ESDS. This means that a SAM ESDS can be:

- Opened via DTF
- Accessed via the SAM imperative macros
- Closed via DTF.

Support is also provided for DTFPH and EXCP access. (In this case, it is space management support only.) The data formats that can be written and read by the EXCP program are entirely under control of the EXCP program itself.

Dynamic secondary allocation is supported according to the access method or EXCP program's constraints and the allocation sizes for the file contained in the VSAM catalog.

Differences between Managed-SAM Access and (Unmanaged) SAM Access

DEFINE considerations:

- RECORDSIZE — If the SAM ESDS is defined *explicitly*, the maximum RECORDSIZE parameter is an important consideration. At open, this maximum record size is compared to the BLKSIZE parameter in the DTF (for TYPEFLE=OUTPUT or WORK only). If the DTF BLKSIZE (minus eight for output) is greater than the maximum record size in the file's catalog entry, VSAM denies the open and cancels the job. VSAM does not allow you to write a larger SAM logical block than the maximum VSAM record size specified during definition. If the SAM ESDS is defined *implicitly*, the maximum VSAM record size for define is determined from the DTF BLKSIZE parameter so that the file's catalog entry is consistent.

  If the DLBL RECORDS/RECSIZE parameters are used for allocation parameters (during implicit define), the RECSIZE parameter specifies the average record size in the file's catalog entry. VSAM DEFINE does not allow the average record size (from the DLBL RECSIZE parameter) to be greater than the maximum record size (from the DTF BLKSIZE parameter). Therefore, if the average record size is larger than the maximum record size, VSAM implicit define sets the average RECSIZE equal to the maximum RECSIZE and the RECORDS value is increased by the same factor that the RECSIZE was decreased by.

- RECORDFORMAT — The record format specified during the explicit define of a SAM ESDS need not match the DTF RECFORM specification. The only exception is that VSAM does not allow a non DTFSD to access a NOCIFORMAT SAM ESDS. Either the file is implicitly redefined or the job is canceled.

- VOLUMES — The VOLUMES specified during define determine the candidates eligible for file allocation. If EXTENT statements are specified with symbolic units during access to the file, there must be an EXTENT statement and a corresponding ASSGN statement for each volume specified during define, even if all of the volumes are not written to or read from. However, EXTENT statements with symbolic units are not required and should not be specified.

- SHAREOPTIONS — The SHAREOPTIONS specified during define affect the sharing characteristics of the file. For example, for a SHAREOPTIONS (1) (one output or many input users) SAM ESDS, if you are updating a file and then want to open the file for input to read it back in, you must close the file. Otherwise, the INPUT OPEN will be denied due to the SHAREOPTIONS specification.

  If you want to allow multiple INPUT with UPDATE users to access a SAM ESDS (in conjunction with the DTF HOLD=YES parameter, for example), explicitly define the file with SHAREOPTIONS (3) to allow concurrent opens for update (that is, OUTPUT). VSAM does not support SHAREOPTIONS (4) for an ESDS. (It is treated as SHAREOPTIONS (2) during OPEN).

  Regardless of the SHAREOPTIONS specification, if you have a file open for load or extension, all other attempts to open that file are denied for reasons of data integrity. Conversely, you would be denied access if you attempted to open a file for load or extension and another user had already opened that file. (A SEOF does not normally exist until a file is closed. If concurrent access was not denied there would be a chance that an input user would read past the end of the file.)

Access Considerations:

- Considerations for all types of files — Many of the considerations introduced with (unmanaged) SAM on FBA devices due to CI format are also considerations in managed-SAM access (on both FBA and CKD devices). For example, if an I/O error occurs during access, it concerns an entire CI of information rather than a single logical block. Also, logical blocks are not necessarily written to a device until a CI is full (compare with the DTF PWRITE parameter).
- VSAM does not distinguish between a file that has just been defined (empty and never written to), a file that has been opened and closed with no records written into it, and a file that has been loaded but deallocated or reset at close. A file in any of these states is considered empty (that is, the high-used RBA is zero). Therefore, in any of these cases, if the file is opened for input via DTFSD, the open will be successful and you will be sent to the EOFADDR on the first GET. This DTF OPEN is actually simulated since VSAM OPEN (ACB) will not open an empty file for input. However, this is transparent to the DTFSD user. Message 4228I with return code X'6E' will be issued to the console by VSAM but this is issued for information purposes only in this case. However, if DTFPH is opened for INPUT on an empty file, VSAM cannot simulate the end of file condition since you do an EXCP before checking for the EOF. This open cannot be allowed because the file has not been opened by VSAM and the file extents have not been located. (The file may not even be allocated in the case of a dynamic file.) Therefore, OPEN DTFPH will be canceled in this situation.
- If EXTENT statements with symbolic units and ASSGN statements are used, and if any one (or more) of the assignments is ignored (IGN), then the entire file is ignored. That is, the DTF is not opened and DTF+X'10', bit 2 (X'20') will be set.
- In general, you should attempt to be as DASD-independent as possible. You should make no assumptions about the track size (or CI size), the size or the number of extents or even the number of volumes that the file will reside on. You should not attempt to choose a BLKSIZE that will maximize DASD utilization because CI format is used and also, your program cannot know what DASD device type will be used for the file before OPEN. It is better to use a smaller BLKSIZE that will be reasonable for any DASD device type in order to be able to process on any DASD device type. You should not assume that a particular symbolic unit will be used. This will allow you to take advantage of VSAM's job control simplifications. Note that (unmanaged) SAM now provides for DASD independence by ignoring the DTF DEVICE= parameter during open.
- Sufficient GETVIS space must be provided for managed-SAM access via use of the SIZE parameter of the EXEC job control statement or the SIZE job control command. The partition GETVIS area must contain at least 40K for the VSAM catalog, plus 10K for each SAM ESDS, plus storage for the CI buffer for each SAM ESDS (see *VSE/VSAM Programmer's Reference* for more information on storage requirements).
- For Work Files:
  - The format of NOTE/POINT IDs for a managed-SAM CKD file is similar to unmanaged SAM FBA NOTE/POINT ID format. That is, for all devices, the managed-SAM NOTE/POINT ID format is CCCN rather than (as for unmanaged SAM) CCHR for CKD and CCCN for FBA. Therefore, you should not generate or modify a NOTE/POINT ID. Also, do not move or modify the DTF between open and close.
  - The DELETFL=NO parameter of DTFSD TYPEFLE=WORK is determined at open. Modifying this indicator after open will have no effect on the close disposition. Note that DLBL DISP specification overrides the DTF DELETFL indicator. If there are any other DTFs or ACBs currently open for this file at

close, the file is not deleted. If the DTF is not closed by the end of job step, automatic CLOSE attempts to close the file.

— Files accessed via DTFSD TYPEFLE=WORK are normally reset at open. If you wish to read a file using a work file DTF, specify DISP=OLD on the DLBL to avoid losing the data due to reset.

Restrictions on Using SAM ESDSs

The following device dependent SAM functions are not supported:

- Split cylinders
- FEOVD (ignored)
- CNTRL (ignored)
- Subsetting of the input file via EXTENT statement specifications.

Certain restrictions apply to specifications on the DTF:

- User labels are not supported. The LABADDR specification of the DTF is ignored.
- CISIZE or BLKSIZE override of the DTF is not supported on the VSAM DLBL job control statement (although the VSAM catalog entry CISIZE does override the DTF).
- Because managed-SAM records are in CI format, SAM spanned records are not supported. VSAM maximum record size (32K minus 7) is not limited by the device track size. The job is canceled if RECFORM=SPNUNB or RECFORM=SPNBLK is specified on the DTF.

There are certain restrictions for SAM ESDS files:

- Files are limited to 16 extents per volume, unless they are explicitly defined as non-reusable (NOREUSE).
- DSF (data secured file) is not supported (it is ignored) on the VSAM DLBL control statement. (VSAM password protection may be used.)

Restrictions concerning the use of DTFPH:

- A file created with DTFPH with CISIZE=0 is not supported by managed-SAM request macros (GET,PUT, etc.). That is, the file can only be read with DTFPH and EXCP. The managed-SAM request macro routines support CI format only. The job is canceled if a non-DTFPH OPEN is issued against a NOCIFORMAT SAM ESDS (unless the file can be implicitly deleted and defined by OPEN).
- Conversely, if a file is created with DTFSD and is to be read via DTFPH with EXCP, the EXCP routine must support it in CI format. If the DTFPH is a version 3 DTF, open stores the CISIZE in the version 3 extension and the open is successful. You must reference this CISIZE when you read the file. If the DTFPH is not a version 3 DTF, open has no means of indicating the CISIZE and the job is canceled.
- There is no way to restrict a DTFPH EXCP user from opening a password-protected SAM ESDS for input (requiring a read password) and then writing to the file. A DTFPH user may also violate SHAREOPTIONS integrity protection in this same manner.

Restrictions on portability of SAM ESDSs:

- SAM ESDSs are not portable and cannot be imported (via IMPORT or IMPORTRA) to OS/VS SAM or VSAM.

IJSYSxx file restrictions:

- The only system data file that is supported is SYSLNK (IJSYSLN). The job is canceled if any other system data files are specified at open.
- System work files (IJSYSnn) are supported unless restricted by the program accessing the system work file.

Note: Some system programs or program products may have restrictions on the use of managed-SAM files. (For example, they may be limited to a single extent or in some cases, managed-SAM files may not be supported.) Please consult the appropriate VSE or Program Product publication for planning and support considerations.

VSAM Access Considerations

VSAM access of a SAM ESDS processes SAM logical records. It uses the RECORDFORMAT information in the catalog to block SAM logical records into SAM logical blocks and de-block SAM logical blocks into SAM logical records. Therefore, it is important that the RECORDFORMAT information in the catalog matches the actual SAM record format of the data.

The valid SAM logical record formats are:

- Fixed unblocked
- Fixed blocked (logicalrecordsize)
- Variable unblocked
- Variable blocked
- Undefined.

SAM access (via DTF) of V or VB records returns the RL (record length field) at the beginning of the record. VSAM access (via ACB) does not return it. Correspondingly, for a PUT for V or VB records, no RL should be at the beginning of the record when it is passed to VSAM because VSAM prefixes the RL. A program using VSAM access (via ACB) for sequential processing can process a VSAM entry-sequenced file or a SAM ESDS.

Differences between the VSAM access of a VSAM entry-sequenced file and the VSAM access of a SAM ESDS are:

- VSAM always loads and extends a SAM ESDS in SPEED mode.
- VSAM does not build an alternate index over a SAM ESDS.
- VSAM does not support path entries over a SAM ESDS.
- VSAM does not support VSAM SPANNED records for a SAM ESDS.

The following applies to VSAM access of a SAM ESDS:

- TCLOSE and ENDREQ do not imply TRUNC or RELSE. VSAM continues processing from the last SAM logical record.
- For direct requests or POINT, the ARG parameter of the RPL always specifies the RBA (relative byte address) of the SAM logical record. The RL and the BL (block length) fields are not included; however, they are accounted for by VSAM. On a direct retrieval, you must supply to VSAM the same RBA as returned during a VSAM load of a SAM ESDS.
- For both the RPL and control block manipulation macros, RECLEN is always the SAM logical record length. It is not the SAM block length (that is, VSAM record length), and does not include any RL or BL fields.
- When issuing a SHOWCB or TESTCB macro, the following apply:
  — NLOGR refers to logical records. For blocked record files, NLOGR could be greater than the number of VSAM records.
  — NRETR and NUPDR have similar meanings; they are the number of retrieved and updated SAM logical records, respectively.

When a SAM ESDS is extended via managed-SAM access, managed-SAM always starts with a new CI. When a SAM ESDS is extended via VSAM access, VSAM attempts to continue storing records into the last CI of the file. Additionally, the number and method of blocking records in a SAM logical block may differ between managed-SAM access and VSAM access, but this is transparent to the sequential access program as long as the program does not reference RBAs.

Access Method Services for a SAM ESDS

This section describes all command parameters with special considerations that apply to VSE/VSAM Space Management for SAM Feature. The commands are listed in alphabetical order. Refer to *Using VSE/VSAM Commands and Macros* for the complete set of Access Method Services parameters.

ALTER command

entryname[/password]
[BUFFERSPACE(size)]
[ERASE|NOERASE]
[EXCEPTIONEXIT(mname)]
[WRITECHECK|NOWRITECHECK]

The ALTER command is used to change attributes in catalog entries. The subparameters and their use for a SAM ESDS are explained here.

entryname[/password]
: is a required parameter that names the SAM ESDS to be altered and supplies its master password if it is password-protected. For a NOCIFORMAT SAM ESDS, the specification of BUFFERSPACE, ERASE, EXCEPTIONEXIT (except as a subparameter of NULLIFY), or WRITECHECK causes the ALTER command to terminate.

[BUFFERSPACE(size)]
: specifies the minimum space to be provided for buffers. For a NOCIFORMAT SAM ESDS, the specification of BUFFERSPACE causes the ALTER command to terminate.

[ERASE | NOERASE]
: specifies whether the SAM ESDS is to be erased when its entry in the catalog is deleted. For a NOCIFORMAT SAM ESDS, the specification of ERASE causes the ALTER command to terminate.

[EXCEPTIONEXIT(mname)]
: specifies the name of the user module to be given control when an exception occurs during the processing of the SAM ESDS whose entry is being altered. For a NOCIFORMAT SAM ESDS, the specification of EXCEPTIONEXIT causes the ALTER command to terminate. (EXCEPTIONEXIT can be specified as a subparameter of NULLIFY.)

[WRITECHECK | NOWRITECHECK]
: specifies whether to check the data transfer of records written in the SAM ESDS via VSAM (ACB) access. For a NOCIFORMAT SAM ESDS, the specification of WRITECHECK causes the ALTER command to terminate.

DEFINE CLUSTER Command

See the section "Explicit Define via the DEFINE CLUSTER Command" for the applicable DEFINE CLUSTER parameters.

DELETE Command

There is no change to the existing DELETE command parameters except that the ERASE parameter is not valid for a NOCIFORMAT SAM ESDS. An implicitly defined

SAM ESDS may be deleted via the DELETE command in the same manner as an explicitly defined SAM ESDS.

EXPORT Command

If you are exporting a CI-format SAM ESDS, VSAM treats it as an entry-sequenced file. If you attempt to export a NOCIFORMAT SAM ESDS, VSAM issues an error message and terminates the command.

You cannot use a SAM ESDS as the portable file (OUTFILE parameter).

EXPORTRA Command

If you are exporting a CI-format SAM ESDS, VSAM treats it as an entry-sequenced file. If you attempt to export a NOCIFORMAT SAM ESDS, explicitly (via the CRA/CRAVOLUMES ENTRIES parameter) or implicitly (via the CRA/CRAVOLUMES ALL parameter), VSAM issues a message, bypasses the file, and continues exporting other requested entries.

You cannot use a SAM ESDS as the portable file (OUTFILE parameter).

IMPORT Command

IMPORT provides full import support for those SAM ESDS files which can be exported. When attempting to import a SAM ESDS into a predefined empty file, IMPORT ensures that the exported file and the predefined file have fully consistent RECORDFORMAT parameter values and that the maximum record size of the predefined file is not less than that of the file originally exported. Any mismatch causes an error message and command termination.

IMPORTRA Command

IMPORTRA provides full import support for those SAM ESDS files which can be exported.

LISTCAT Command

You can display space for a SAM ESDS by specifying LISTCAT SPACE. You can display all files that have been defined for a particular catalog by using the LISTCAT command; this includes all SAM ESDSs defined either explicitly or implicitly.

The ATTRIBUTES portion of LISTCAT output is modified as follows for ESDS:

- CISIZE indicates 0 if RECORDFORMAT(NOCIFORMAT) was specified on either an explicit or implicit define.
- CI/CA indicates 0 if RECORDFORMAT was specified on either an explicit or implicit define.
- SAMLRECL indicates the SAM logical record length (listed for SAM ESDS only). This value is the user-supplied record length for FIXBLK SAM files and is zero for all other record format SAM files.
- RECFORMAT indicates the SAM record format (listed for SAM ESDS only). FIXBLK, FIXUNBLK, VARBLK, VARUNBLK, UNDEFINED, or NOCIFMT are the possible values for this attribute.
- IMP-DEFINE is listed if the SAM ESDS has been implicitly defined; otherwise EXP-DEFINE is listed (applies to SAM ESDSs only).
- SAMDATASET is listed if the ESDS is a managed-SAM file; otherwise VSAMDATASET is listed.

The Statistics Group (data) is listed for a SAM ESDS. However, it should be noted that these statistics are maintained during VSAM access only and not during managed-SAM access.

PRINT Command

You can print a CI-format SAM ESDS via either managed-SAM access or VSAM access. The output is always SAM logical records. You cannot print a NOCIFORMAT ESDS via either managed-SAM or VSAM access.

For managed-SAM access:

- Specify the ENVIRONMENT parameter.
- The output format is the same as unmanaged SAM (no RBA display, record length field at the beginning of the record for format V and VB).
- The SKIP and COUNT parameters can be used and the value always indicates the number of SAM logical records to be skipped or listed.

(See Example 4 in Appendix A for an example of printing a SAM ESDS by retrieving the SAM logical records with managed-SAM.)

For VSAM access:

- Omit the ENVIRONMENT parameter.
- The output format is a VSAM entry-sequenced file (RBA display, no record length field at the beginning of format V and VB RECORDS. VSAM uses the RECORDFORMAT information recorded in the catalog to determine the SAM record format for access).
- The SKIP and COUNT parameters can be used and the value always indicates the number of SAM logical records to be skipped or listed.
- The FROMADDRESS and TOADDRESS parameters can be used. (The RBA value for FROMADDRESS must be the exact beginning of a SAM logical record.)

REPRO Command

CI-format SAM ESDSs can be used as input or output files in a REPRO command wherever SAM files or VSAM entry-sequenced files are currently allowed. (Do not specify a NOCIFORMAT SAM ESDS as an input or output file.) You can use the REPRO command to convert an unmanaged SAM file to a SAM ESDS.

- INFILE(dname[/password][ENVIRONMENT(subparameters)]) indicates the CI-format SAM ESDS to be used as the input file.
- OUTFILE(dname[/password][ENVIRONMENT(subparameters)]) indicates the CI-format SAM ESDS to be used as the output file. If the output file is a managed-SAM file that is to be created via managed-SAM access, and it has not been previously defined, it will be implicitly defined if the job control statements meet the requirements of implicit define.
  - For both the INFILE and OUTFILE parameters, *dname* specifies the *filename* of the DLBL job control statement that identifies the file to be copied. You may specify the ENVIRONMENT parameter for managed-SAM access. SAM logical records are always the unit of output. *password* is not allowed for managed-SAM access.
- FROMADDRESS(address) TOADDRESS(address)
  You can specify FROMADDRESS and TOADDRESS for VSAM access (not managed-SAM access). The RBA value for FROMADDRESS must be the exact beginning of a SAM logical record.
- SKIP(count) COUNT(count)
  You can specify SKIP and COUNT (for both VSAM and managed-SAM access) and the value always indicates the number of SAM logical records to be skipped or copied.

VERIFY Command

If the VERIFY command is executed on a CI-format SAM ESDS, you can discover whether the file was successfully closed (warning messages are issued), but you cannot cause the end-of-file indicator in the catalog entry to be updated. This is because a SAM ESDS is always loaded and extended in SPEED mode. A SAM ESDS cannot be accessed for input by VSAM unless it was successfully closed after being initially loaded. (If the file is accessed for input by managed-SAM without being closed, an open in a subsequent job step will be successful and the first GET will cause the user to be sent to the EOFADDR routine.) The file can only be accessed by VSAM up to the data written by the last successful close if extended. After extension, a SAM ESDS is able to be accessed by managed-SAM even if the close was unsuccessful; however, the file may not terminate with an SEOF.

The VERIFY command terminates due to an open error if it is executed on a NOCIFORMAT SAM ESDS.

LISTCRA Command

Use the LISTCRA command to validate and recover a catalog with SAM ESDS entries. No special considerations apply for a SAM ESDS.

RESETCAT Command

Use the RESETCAT command to validate and recover a catalog with SAM ESDS entries. No special considerations apply for a SAM ESDS.

Deleting a SAM ESDS

Explicit Delete - DELETE Access Method Services Command

There is no change in the use of the existing DELETE command parameters to delete a SAM ESDS (except that the ERASE parameter is not valid for a NOCIFORMAT SAM ESDS).

Implicit Delete

An implicit delete of a SAM ESDS occurs if all the following conditions are true for any of the following cases:

Case 1: During open of DTF (implicit delete followed by implicit define)

- The catalog entry has been implicitly defined.
- The DTFSD maximum logical blocksize exceeds the VSAM catalog maximum RECORDSIZE of the SAM ESDS or the RECORDFORMAT of the file is NOCIFORMAT.
- DTFSD TYPEFLE=OUTPUT or WORK.
- The file is unexpired and the operator has responded "delete" to message 4233A EQUAL FILE-ID IN CATALOG, or the file is expired.
- DISP=OLD is not specified.

Case 2: During close of DTF

- The catalog entry has been implicitly defined.
- DISP=(...,DELETE) or DTF TYPEFLE=WORK and DELETFL is unspecified. (**Note:** The job control statement overrides the DTF.)

Case 3: During close of DTF

- The catalog entry has been implicitly defined.
- DISP=(...,DATE) is specified.
  (Note: The job control statement overrides the DTF.)
- The expiration date has passed.

For all of the cases, if another user has the same file open for access, the file will not be deleted.

Appendix A: Sample Programs and Job Streams

Example 1: Load a SAM ESDS via Managed-SAM Access

A program that loads a SAM ESDS via managed-SAM access (source code). Examples 2, 3, and 4 use this program assuming that it is cataloged under the phase name SDOUTPUT.

```
Col.        Col.    Col.                                                Col.
1           10      16                                                  72
SDOUTPUT    START   X'200078'
            BALR    2,0
            USING   *,2
            OPEN    SDOUT,PRINT
            LA      5,1                 INITIAL COUNT TO 1
            L       6,MAXRCDS           LOAD NO. OF RECORDS TO WRITE
LOOP        CR      5,6                 WRITTEN LAST RECORD YET?
            BH      CLOSE               YES
STORE       ST      5,RECNO             NO, STORE RECORD NUMBER
            CVD     5,DWB               CONVERT KEY TO DECIMAL
            UNPK    NUM(15),DWB(8)      UNPACK KEY
            TM      UNPKSIGN,X'10'       SEE IF NUMBER WAS NEGATIVE
            BO      NEG1                 YES, NEGATIVE
            MVI     SIGN,C'+'            MAKE OUTPUT SHOW POSITIVE
            B       CONTINUE
NEG1        MVI     SIGN,C'-'            MAKE OUTPUT SHOW NEGATIVE
CONTINUE    OI      UNPKSIGN,X'F0'       MAKE LAST BYTE A NUMBER
            PUT     PRINT               PRINT KEY
            PUT     SDOUT,WORKAREA      PUT FROM WORKAREA
            LA      5,1(5)              INCR RECORD NO.
            B       LOOP                GO BACK
CLOSE       CLOSE   SDOUT,PRINT         CLOSE THE FILE
            EOJ
            EJECT
*
SDOUT       DTFSD   BLKSIZE=2008,1                                      X
                    DEVADDR=SYS007,2                                    X
                    IOAREA1=OUTPUT1,                                    X
                    DEVICE=2314,3                                       X
                    RECFORM=FIXBLK,4                                    X
                    RECSIZE=80,                                         X
                    TYPEFLE=OUTPUT,5                                    X
                    WORKA=YES
            EJECT
PRINT       DTFDI   DEVADDR=SYSLST,                                     X
                    IOAREA1=IOAREA,                                     X
                    RECSIZE=17
*
            EJECT
DWB         DC      D'0'                USED TO CVD INTO
*
IOAREA      DC      0CL17' '
            DC      C' '                PRINT CONTROL
OUT         DC      0CL16' '
SIGN        DC      C' '                PRINTED SIGN
NUM         DC      0CL15' '            PRINTED KEY
            DC      CL14' '
UNPKSIGN    DC      C' '                LAST BYTE OF UNPACKED NUMBER
*
MAXRCDS     DC      F'400'                  NO. OF RECORDS TO WRITE
*
WORKAREA    DC      0CL80' '
RECNO       DC      F'0'                    CURRENT RECORD NO.
            DC      CL76' '
OUTPUT1     DC      CL8' '                  AREA FOR COUNT
            DC      25CL80' '
            EJECT
*           SDMODFO6                                                    X
            PRINT NOGEN
            DIMOD TYPEFLE=OUTPUT
            END SDOUTPUT
```

Explanations for Example 1:

1. The BLKSIZE specifies the logical blocksize of the SAM file. The extra eight bytes specified include the count area required for DTFSD OUTPUT data files.
2. This symbolic unit is ignored. The symbolic unit either comes from the EXTENT statement or is dynamically chosen by VSAM.
3. The device type specified is ignored. VSAM determines the device type from the volume serial of the volume that the file resides on. The volume serial is specified either in the EXTENT statement during implicit define, or in the VOLUMES parameter of the DEFINE CLUSTER command, or is chosen by VSAM from a default model during explicit or implicit define.
4. The RECFORM along with the BLKSIZE and RECSIZE information is used to determine the record format and size characteristics of the file to be written. In addition, if the file is implicitly defined, this information is stored into the VSAM catalog to be used if the file is accessed via VSAM (ACB).
5. An output file normally implies reset. That is, the file is set to empty before the records are written into the file. This may be overridden by the DLBL DISP parameter. If DISP=OLD is specified, the file will not be reset and an existing file will have this data added to the end of the file. (If the file does not exist or is empty, DISP=OLD or NEW has no effect.)
6. No SD logic module needs to be assembled or included. (Note that it is a comment.)

Note: If IOAREA2 is specified in the DTFSD (in combination with either IOREG or WORKA) and implicit define occurs, VSAM will attempt to choose a CI size that will hold at least two SAM logical blocks.

Example 2: Implicit Define of a SAM ESDS.

A job that loads a SAM ESDS via managed-SAM access (execution). This job implicitly defines a SAM ESDS.

```
// JOB LOAD A MANAGED SAM FILE (400 RECORDS)
// DLBL SDOUT,'MANAGED.SAM.FILE1',0,VSAM,RECORDS=400,RECSIZE=80[1]
// EXTENT ,VSER01[1]
// EXEC SDOUTPUT,SIZE=AUTO[2]
/&
```

Explanations for Example 2:

1. The information from the DLBL and the EXTENT statement (together with the DTF information) provides the information to do an implicit define.
   - — The DLBL specifies VSAM indicating that the SAM file is to be a SAM ESDS.
   - — A retention period of 0 indicates that the file can be deleted at any time (assuming it is not in use).
   - — RECORDS and RECSIZE specify that the primary allocation size should be large enough to hold four hundred records of eighty bytes each. The secondary allocation size is assumed to be twenty percent of the primary allocation size.
   - — The default disposition for an OUTPUT DTFSD data file is (NEW,KEEP).
   - — The EXTENT statement specifies that the file is to reside on volume VSER01 and that the logical unit is to be dynamically assigned by VSAM. (This assumes, of course, that there is available VSAM data space on volume VSER01.)
2. A SAM ESDS needs about 52K of GETVIS space for access (12K for the file and CI buffers, and a one-time requirement of 40K for the catalog). SIZE=AUTO will ensure that the maximum GETVIS space is available to VSAM.

Example 3: Define a Default Model SAM ESDS

A job stream that loads a SAM ESDS via SAM access (execution). This job stream defines a default model for a SAM ESDS and then implicitly defines a SAM ESDS, using the default model to obtain a volume list (to allow elimination of the EXTENT statement).

```
// JOB DEFINE DEFAULT MODEL FOR SAM ESDS
// EXEC IDCAMS,SIZE=AUTO
                DEFINE CLUSTER -
                 (NAME(DEFAULT.MODEL.ESDS.SAM) -1
                  VOLUMES(VSER02) -2
                  RECORDS(100 25) -
                  RECORDSIZE(2000 2000) -
                  RECORDFORMAT(UNDEF) -
                  REUSE -
                  NOALLOCATION -3
                  NONINDEXED)
                LISTCAT -
                  ENTRIES(DEFAULT.MODEL.ESDS.SAM) -
                  ALL
/*
/&
// JOB LOAD A MANAGED SAM FILE (400 RECORDS)
// DLBL SDOUT,'MANAGED.SAM.FILE2',0,VSAM,RECORDS=400,RECSIZE=804
// EXEC SDOUTPUT,SIZE=AUTO
/&
```

Explanations for Example 3:

1. This is the required file-ID for a default model for a SAM ESDS.
2. This is the volume that will be used for any SAM ESDS that is implicitly defined with no EXTENT statement specified (or explicitly defined with no VOLUMES parameter specified).
3. NOALLOCATION is required for default model.
4. The same DLBL information is specified as in Example 2, but the volume that the file is to reside on is retrieved from the default model rather than an EXTENT statement. (The file will reside on VSER02.) Also, the symbolic unit is dynamically chosen and assigned by VSAM. Note that allocation size and retention period are still obtained from the DLBL statement. The only information retrieved from the default model during implicit define is the volume list.

Example 4: Define a Dynamic SAM ESDS and Access.
A job stream that loads a SAM ESDS via managed-SAM access (execution). This job stream defines a dynamic SAM ESDS and then accesses the defined file allowing elimination of the EXTENT statement.

```
// JOB ESDS DEFINE FOR SAM ESDS
// EXEC IDCAMS,SIZE=AUTO
                DEFINE CLUSTER -
                 (NAME(MANAGED.SAM.FILE3) -
                  VOLUMES(VSER03) -
                  RECORDS(16 4) -1
                  RECORDSIZE(2000 2000) -2
                  RECORDFORMAT(FIXBLK(80)) -3
                  REUSE -4
                  NOALLOCATION -
                  NONINDEXED)5
                LISTCAT -
                  ENTRIES(MANAGED.SAM.FILE3) -
                  ALL
/*
/&
// JOB LOAD A MANAGED SAM FILE (400 RECORDS)
// DLBL SDOUT,'MANAGED.SAM.FILE3',,VSAM,DISP=(NEW,KEEP)6
// EXEC SDOUTPUT,SIZE=AUTO
/&
// JOB EXTEND A MANAGED SAM FILE (ANOTHER 400 RECORDS)
// DLBL SDOUT,'MANAGED.SAM.FILE3',,VSAM,DISP=(OLD,KEEP)7
// EXEC SDOUTPUT,SIZE=AUTO
/&
// JOB ESDS PRINT A MANAGED FILE WITH SAM
// DLBL ESD1,'MANAGED.SAM.FILE3',,VSAM,DISP=(OLD,DELETE)8
// EXEC IDCAMS,SIZE=AUTO
                PRINT INFILE(ESDS1 -
                  ENVIRONMENT -9
                       (BLOCKSIZE(2000) -
                       RECORDFORMAT(FIXBLK) -
                       RECORDSIZE(80)))
/*
/&
```

Explanations for Example 4:

1. This specifies the number of VSAM logical records (SAM logical blocks) for primary and secondary allocation. Sixteen is specified for the primary allocation since sixteen 2000-byte logical blocks will be written to hold four hundred 80-byte SAM logical records.
2. This specifies the average and maximum VSAM logical record size (SAM logical block size) for the file.
3. This specifies the SAM logical record size for the file.
4. REUSE in connection with NOALLOCATION makes the file a dynamic file.
5. NONINDEXED is required for a SAM ESDS.
6. The DISP parameter specifies that the file is to be reset at open, and kept at close. The file resides on volume VSER03 as specified in the define cluster. A symbolic unit will be dynamically assigned.
7. The DISP parameter specifies that the file is *not* to be reset at open, that is, the file will be extended with the records written by JOB EXTEND. The file is to be kept at close.
8. The DISP parameter specifies that the file is *not* to be reset at open. (A specification of DISP=NEW would be an error in this case since the file will be opened for input.) When the file is closed, it will be deleted (that is, deallocated since this file was defined as a dynamic file).
9. The file is read via managed-SAM access by way of the ENVIRONMENT parameter. The ENVIRONMENT subparameters specify the information required to generate a DTF. The file may be accessed via VSAM by omitting the

ENVIRONMENT parameter. In that case, VSAM gets the SAM file characteristics from the VSAM catalog entry for the file.

Appendix B: Differences between (VSAM) ESDS and SAM ESDS Format

Figures 6 and 7 illustrate the way in which control intervals are physically formatted into control areas for VSAM entry-sequenced files as opposed to SAM ESDSs.
Assumptions:

Device type=3340
Allocation specified=TRK (3 1)
CI size=2K
Physical block size=1K
1 track=7 physical records (PR)
11 CIs of data are written
CA=Min (primary [3 TRKs], secondary [1 TRK], Max-CA[1 CYL])

Therefore: CA=1 track

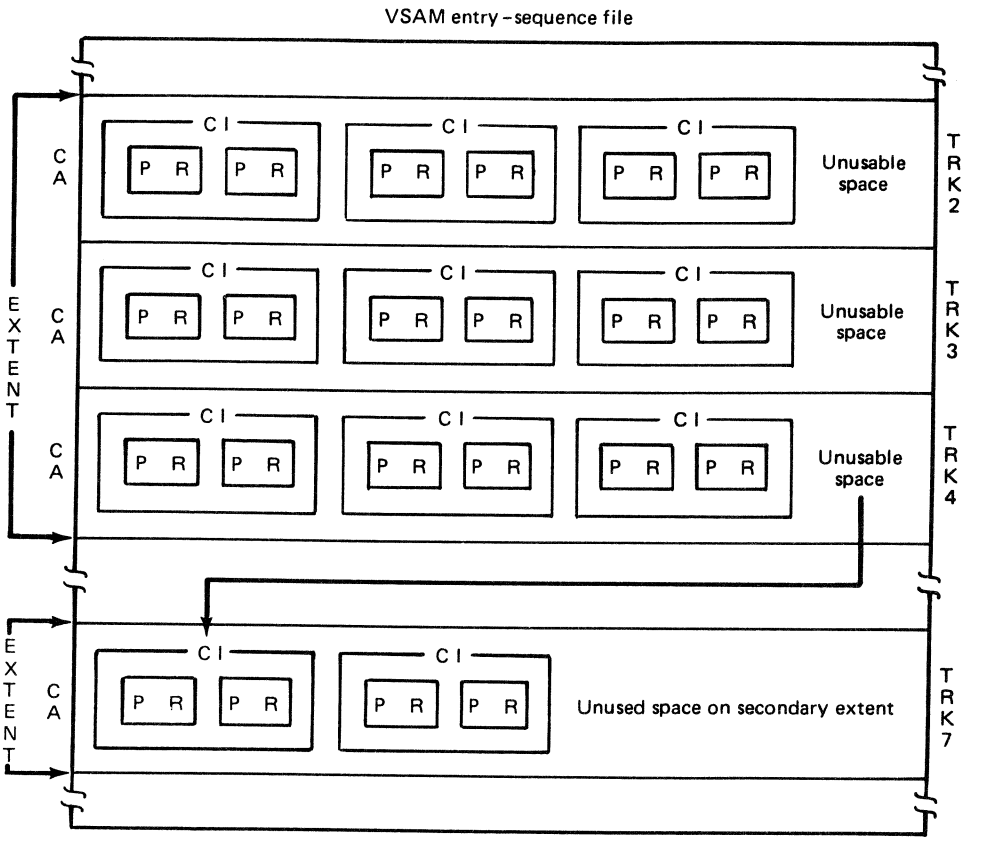

Figure 6. Example of CA Format Using a VSAM Entry-Sequenced File

A VSAM entry-sequenced file formats CIs into CAs in CA format. This means that CIs cannot be written across CA boundaries. If there is not sufficient space at the end of a CA to write a complete CI, an area of unusable space is left and the CI to be formatted is written at the beginning of the next CA.

Assumptions:

Device type=3340
Allocation specified=TRK (3 1)
CI size=2K
Physical block size=1K
1 track=7 physical records (PR)

11 CIs of data are written
CA=Min (primary [3 TRKs], secondary [1 TRK], Max-CA[1 CYL])

Therefore: CA=1 track

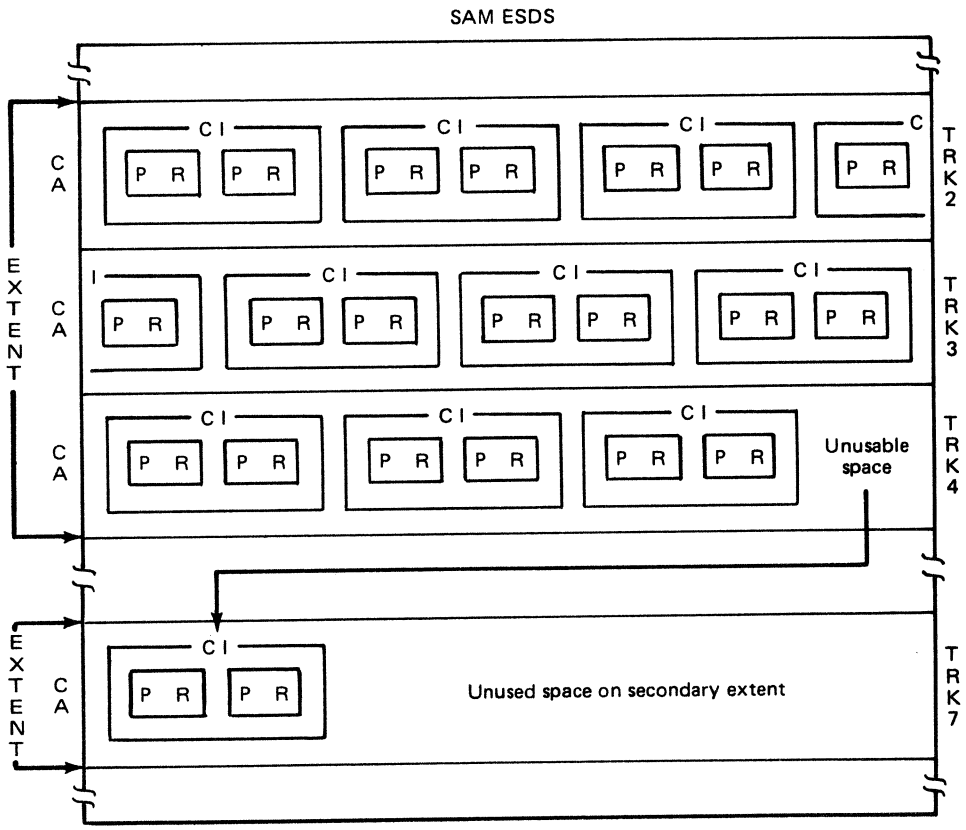

Figure 7. Example of Non-CA Format Using a SAM ESDS

A SAM ESDS formats CIs into CAs in non-CA format. This means that a CI is able to be written across a CA boundary (tracks 2 and 3) but not across an extent boundary. If there is not sufficient space at the end of the CA to write a complete CI, the CI will be written across a CA boundary causing the CI to have part of its contents in one CA and the rest of its contents in another.

Figure 8 shows the construction of a control interval and how records are physically and logically laid out for a VSAM entry-sequenced file and a SAM ESDS.

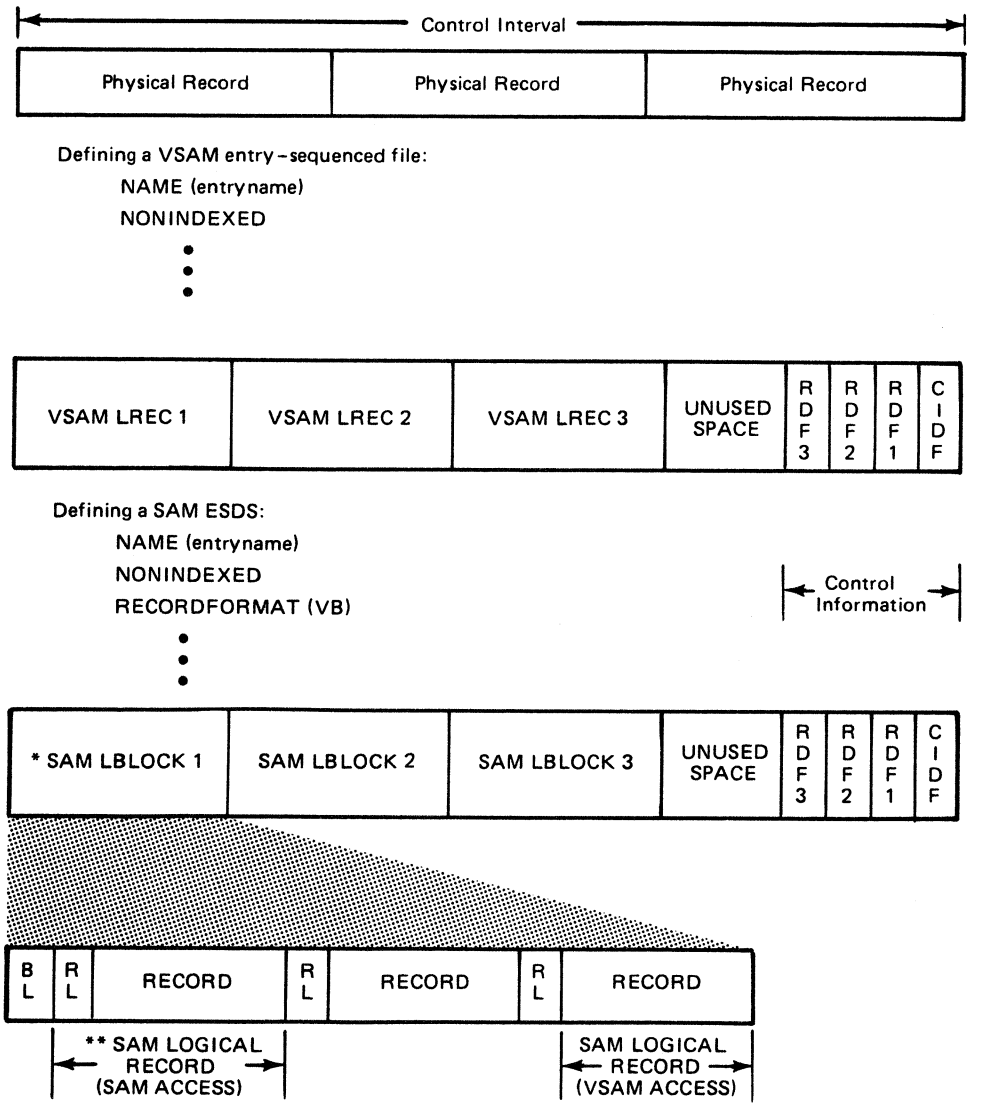

Figure 8. Comparison of a VSAM Logical Record to a SAM Logical Block

* The SAM LOGICAL BLOCK size is what you specify in the RECORDSIZE parameter when you DEFINE a SAM ESDS and in the BLKSIZE parameter of the DTF.

** The SAM LOGICAL RECORD size is what you specify in the RECORDFORMAT parameter when you DEFINE a FIXBLK SAM ESDS and in the RECSIZE parameter of the DTF.

When you have defined a VSAM entry-sequenced file, the control interval will be made up of VSAM logical records and their related control information. When you define a SAM ESDS, the VSAM logical records become SAM logical blocks. The control interval size is a multiple of the VSAM physical record size and is normally determined by VSAM, not you, at DEFINE time. Control information in a control interval consists of a CIDF (Control Interval Definition Field) and RDFs (Record Definition Fields). There is an RDF for each SAM logical block (VSAM logical

record) indicating its length, except in the case of consecutive logical blocks of equal length, in which case the first RDF (right-most of the pair) describes the length of the logical blocks and the second RDF (left-most of the pair) tells how many logical blocks the first RDF describes.

The SAM logical block consists of SAM logical records. In the case of VB format, each logical record is prefixed with an RL (record length) field which indicates the length of the record. The SAM logical block begins with a BL (block length) field which indicates the length of the block. During managed-SAM (DTF) access of V or VB records, the RL is returned at the beginning of the record. For VSAM (ACB) access, it is not. A program using VSAM (ACB) access for sequential processing can process a SAM ESDS or a VSAM entry-sequenced file.

Glossary

allocation. The assignment of a resource (in this case, DASD space) for use in performing a specific task. VSAM always allocates an integral number of CAs.

BL. (See block length.)

block length. The high-order 4-byte field at the beginning of a block of SAM variable-length records which contains the control information consisting of the block length including the length of the control information (block length and record length fields) itself.

CA. (See control area.)

CI. (See control interval.)

CIDF. (See control interval definition field.)

CKD. (See count-key-data.)

control area. In VSAM, a group of control intervals and also a unit of allocation.

control interval. A fixed-length area of direct access storage in which VSAM stores records and distributes free space. It is the unit of information that VSAM transmits to or from direct access storage. A control interval always comprises an integral number of physical records.

control interval definition field. In VSAM, a field located in the four right-most bytes of each control interval; describes the free space (if any) in the control interval.

count-key-data. A direct access storage device that contains a fixed-length count area, a variable-length key area (optional), and a variable-length data area in each of its physical data blocks.

explicit DEFINE/DELETE. Used to place information describing the SAM ESDS into the VSAM catalog, based on parameters specified in the Access Method Services DEFINE CLUSTER command, or to remove all references to the SAM ESDS from the VSAM catalog and release the file's space, based on parameters specified in the Access Method Services DELETE command.

extent. A continuous space on a direct access storage volume, occupied by or reserved for a particular data set, data space, or file. A file extent is always an integral number of CAs.

FBA. (See fixed block architecture.)

fixed block architecture. An architecture describing a direct access storage device that is linear in structure; it contains fixed-length (512 byte) blocks of data.

implicit define/delete. Used to place information describing the SAM ESDS into the VSAM catalog during open of a DTF for OUTPUT or WORK without the use of Access Method Services DEFINE CLUSTER command, or to remove all references to the SAM ESDS from the VSAM catalog during close of a DTF without the use of Access Method Services DELETE commands.

logical block. One or more logical records grouped together. A logical block consists of one unit of data as it would be transferred to a device (physical block) when unmanaged SAM is used on a CKD device. This data entity is called a logical block in the managed-SAM environment instead of a physical block because several of them may be combined into a CI for transfer to the DASD device. The blocking structure as specified by the BLKSIZE DTF parameter is maintained in logical blocks to provide support for programs having dependencies on the CKD physical blocks concept. When working with unblocked records, a logical block contains exactly one logical record.

logical record. (1) A record independent of its physical environment. Portions of the same logical record may be located in different physical records, or several logical records or parts of logical records may be located in one physical record. (2) A record from the standpoint of its content, function, and use rather than its physical attributes; that is, one that is defined in terms of the information it contains. (3) In VSAM, a unit of information normally pertaining to a single subject; a logical record is that user record requested of or given to the data management function.

managed-SAM access. The use of SAM access (via DTF) in combination with VSAM space management facilities. VSAM chooses file locations within VSAM data space(s).

max-CA. (See min-CA.)

min-CA. The terms "tracks" and "cylinders" as used for a CKD (count-key-data) device are not necessarily meaningful for an FBA (fixed block architecture) device because an FBA device stores data in "blocks". FBA uses a linear addressing scheme whereby blocks are not necessarily associated with physical characteristics, such as cylinders or tracks.

Min-CA replaces the former term, "track"; it represents minimum control area size for both CKD and FBA devices. Min-CA and max-CA are units of allocation. The size of these units of allocation depends upon the device being used. CKD devices can have more than one min-CA (track) and max-CA (cylinder). Sizes for CKD devices are dependent upon the number of physical records contained in them (a function of the control interval size selected). An FBA device can have only one min-CA and one max-CA value.

physical record. A physical unit of recording on a medium, for example, the physical unit between address markers on a disk. In VSAM, one or more physical records make up a control interval.

RBA. (See relative byte address.)

RDF. (See record definition field.)

record definition field. In VSAM, a field stored as part of a stored record segment; it contains the control information required to manage stored record segments within a control interval.

record length. The high-order 4-byte field at the beginning of a variable-length record which contains control information consisting of the record length including the length of the control information (the RL field) itself.

relative byte address. The displacement of a data record or a control interval from the beginning of the file to which it belongs, independent of the manner in which it is stored.

RL. (See record length.)

SAM. Sequential Access Method. An access method for storing or retrieving data blocks in a continuous sequence, using a tape or direct access storage device. Managed-SAM access as used in this text only applies to data stored on direct access storage devices.

SAM block. For unmanaged CKD SAM, a group of contiguous characters transmitted in a single operation, to or from a direct access device. For managed-SAM, the SAM block is mapped into a SAM logical block.

SAM ESDS. A SAM file within VSAM data space accessible via a SAM DTF or VSAM ACB and described in the VSAM catalog. It is defined explicitly via the DEFINE CLUSTER command with the RECORDFORMAT parameter, or by an implicit define at DTF OPEN.

SAM logical block. A block of data consisting of SAM records and their associated control information; together they make up a VSAM logical record. One or more is mapped into a control interval.

SAM logical record. For managed-SAM access, the SAM logical record consists of just the SAM record for FIXBLK, FIXUNB and UNDEF records. For VARUNB or VARBLK records it consists of both the record-length (RL) field and the record itself.

SEOF. Software End-Of-File. In VSAM, a code that indicates that no data exists beyond it in the component. A software end-of-file is indicated by a zero value control interval definition field in conjunction with information in the VSAM catalog.

stored record. The combination of a data record and its associated control information, even though they are not necessarily adjacent to each other.

unmanaged SAM access. The use of SAM access in combination with space management by way of the VTOC. The user specifies the location of files during creation via the extent limits on the EXTENT statement(s).

VERIFY command (Access Method Services). Usually used following a system failure to compare the end-of file information in a VSAM catalog with the end-of-file indicator(s) in the file itself. If the information does not agree, the catalog is corrected.

It is not operative on a file that is defined with SPEED and is therefore not generally useful for a SAM ESDS. Not to be confused with the DTF VERIFY parameter for SAM access. (See Using VSE/VSAM Commands and Macros.)

VERIFY parameter (DTF). Checks for correct data transfer during managed-SAM access by specifying DTF VERIFY = YES or NO. Not to be confused with the Access Method Services VERIFY command for VSAM access.

VSAM. Virtual Storage Access Method.

VSAM entry-sequenced file. A file where records are stored in the physical sequence in which they are loaded. Once a record is loaded, its RBA (relative-byte address) never changes. Direct access is possible with a VSAM entry-sequenced file by keeping track of records' RBAs and associating these RBAs with the contents of the records.

VSAM ESDS. (See VSAM entry-sequenced file.)

VSAM logical record. The part of a control interval described by an RDF or a pair of RDFs. (See SAM logical blocks.)

VSAM managed space. VSAM data space.

VSAM physical record. (See physical record.)

Index

access considerations 21
access method service commands 7
access method services for a SAM ESDS 24
access methods and file types, valid combinations 12
accessing data via DTFPH with EXCP 16
accessing files via DTFSD TYPEFLE=WORK 22
additional considerations 16
additional functions available for managed-SAM access 8
allocating space for explicitly defined files 7
allocating space for implicitly defined files 7
allocation
  dynamic secondary 19
  primary 9
  secondary 9
  single extent primary 16
allocation of file space 6
allocation size
  default value 7
  primary 19
  secondary 19
ALTER command 24
alternate index, building over SAM ESDS 16
ASSGN statement 20
assumptions for implicit define 17
assumptions made by VSAM 17
automatic close 8
automatic closing of data files 7
automatic space management 9

BLKSIZE parameter 17, 18, 20
blocks, number of 19
BLOCKS parameter in DEFINE CLUSTER command 15
BUFFERSPACE parameter 24
building an alternate index over a SAM ESDS 16

CAT operand 18
catalog recovery areas
  comparing 8
  listing 8
catalog recovery support 8
CI format
  maximum record size 18
  record format 18
CISIZE parameter 17, 18
CLOSE DTF 10
close of DTF 27, 28
commands, access method services 7, 24-27
comparing catalog recovery areas 8
comparison of a VSAM logical record to a SAM logical block 36
considerations
  access 21
  additional
  define 19, 20
  define and managed-SAM access 19
  VSAM access 23
control interval size 17, 18
controlling access to data files 7
converting SAM files to VSAM files 5
converting to VSAM, advantages 5
creating portable copies 8
CYLINDERS parameter in DEFINE CLUSTER command 15

DASD independence 6, 7
DASD sharing of SAM files 8
data component level 16
data files
  automatic closing 7
  controlling access 7
data recovery 8
data, security and integrity of 7
date 18
default model of a SAM ESDS 19
default modeling 6
default value for allocation size 7
define a default model SAM ESDS 31
define a dynamic SAM ESDS and access 32
define and managed-SAM access considerations 19
DEFINE CLUSTER command 24
  for explicit define 14
DEFINE CLUSTER parameters
  explanations 14
  for defining a SAM ESDS 14
define considerations 19, 20
defining a path over a SAM ESDS 16
defining a SAM ESDS 13
DELETE access method services command 27
DELETE command 24
DELETFL=NO parameter 21
deleting a SAM ESDS 27
determining control interval size 17
determining maximum VSAM logical record length 17
determining record format 17
determining SAM logical record size 17
differences
  between managed-SAM access and SAM access 20
  between (VSAM) ESDS and SAM ESDS format 34
  VSAM entry-sequence file to SAM ESDS 23
DISP on DLBL statement 10
disposition of files at CLOSE 9
disposition of work files 9
disposition parameter specifications 9
DLBL DISP parameter 10
  space allocation control 6
DLBL DISP specification 21
DLBL statement 10
DTF, closing 27, 28
DTF DELETFL indicator 21
DTF information, obtained by VSAM 17
DTF, opening 27
DTF RECSIZE parameter 17
DTF, specification restrictions 22
DTFPH 16
  access 19
  MOUNTED=SINGLE 17, 18
  restrictions for use 22
DTFSD 17
DTFSD specifications 17
  for work files 18
dynamic allocation 6
dynamic file 19
dynamic file capability of work files 16
dynamic primary allocation 9
dynamic secondary allocation 9, 19

ECXP program 19
ERASE parameter 24
examples
  define a default model SAM ESDS 31
  define a dynamic SAM ESDS and access 32
  implicit define of a SAM ESDS 30
  load a SAM ESDS via managed-SAM access 29
  of a CA format using a VSAM entry-sequenced file 34
  of a non-CA format using a SAM ESDS 35
EXCP 16
EXCP access 19
EXECPTIONEXIT command 24
explicit define 13
  the DEFINE CLUSTER command 14
explicit delete 27

EXPORT command 25
EXPORTRA command 25
extending existing SAM files 10
EXTENT statement 19

file definition, implicit 6
file migration 11
file not successfully closed condition, resetting 8
file space, allocating 6
file status 10
file types and access methods, valid combinations 12
file-ID 18
files in REUSE attribute 10
files, work 9

gathering information 17
GETVIS space 21
glossary 38

HOLD=YES specification 8
how to use VSE/VSAM space management for SAM feature 13

ICCF 16, 18
IJSYSxx file restrictions 22
implicit define 13
  assumptions 17
  cluster 16
  information 19
  information for 18
  of a SAM ESDS 30
implicit delete 27
implicit file definition 6
IMPORT command 25
IMPORTRA command 25
independence, DASD 6, 7
information for implicit define 18, 19
information gathering 17
information obtained from the DTF 17
information obtained from the job control statements 18
integrity 7
  of data 7
interactive computing and control facility 16, 18
introduction 5

job control, simplified 6
job control statement information, obtained by VSAM 18
job control statements, restrictions of 6
job streams, sample 29

level of shared access 8
levels of migration 11
levels of security 7
list catalog recovery areas 8
LISTCAT command 25
LISTCRA command 27
load a SAM ESDS via managed-SAM access 29
logical block size 19
  for SAM 36
logical record length, determining 17
logical record size 19
  determining 17
  for SAM 36
managed-SAM access close disposition 10, 11
managed-SAM access open disposition 10
managed-SAM open 16
maximum record size 17, 18
  CI format 18
  non-CI format 18
maximum VSAM logical record length, determining 17
message 4228I 21
migration from SAM to VSAM control 12
migration, levels of 11
migration of files 11
migration steps 11
modeling attributes 16
moving data and programs from SAM to VSAM control 11
moving from SAM to VSAM 11
multiple extents 8
multiple volumes 8
NAME parameter in DEFINE CLUSTER command 14
NOALLOCATION parameter 19
NOCIFORMAT SAM ESDS 20
NOERASE parameter 24
non-CI format
  maximum record size 18
  record format 18
NONINDEXED parameter
  assumptions 17
  in DEFINE CLUSTER command 14
NONSPANNED parameter assumptions 17
NOTE/POINT ID format 21
NOWRITECHECK parameter 24
  assumptions 17
number of blocks 19
number of tracks 19

OPEN DTF 10
OPEN DTFPH 21
open of DTF 27
OPEN options 9
operations required to use space management for SAM feature 13
options at OPEN 9

partition and processor independent file-ID 16
partition independent file-ID 16
partition/processor independence specification 16
partition/processor unique file-ID 18
partition/processor uniqueness 18
partition-unique file-ID, specifying 16
password protection 7
path, defining over a SAM ESDS 16
portable copies, creating 8
preventing unauthorized access 7
primary allocation 9
  size 19
primary space, allocated as single extent 16
PRINT command 26
processing of SAM ESDSs 8
processor independent file-ID 16
processor-unique file-ID, specifying 16
programs, sample 29
protection
  password 7
  security 7
RECFORM parameter 17, 18
record format
  CI format 18
  determining 17
  non-CI format 18
record size 18
RECORDFORMAT
  information in the catalog 23
  parameter 20
  parameter in DEFINE CLUSTER command 14
RECORDS
  operand 18
  parameter 19, 20
RECORDSIZE parameter 20
  in DEFINE CLUSTER command 15
recovery, data 8
RECSIZE operand 19
RECSIZE parameter 19, 20
REPRO command 26
requesting allocation sizes 7
RESETCAT command 27

resetting and reusing a previously-defined file 19
resetting the file not successfully closed condition 8
restrictions
  for IJSYSxx file 22
  for SAM ESDS files 22
  for use of DTFPH 22
  of job control statements 6
  of portability of SAM ESDSs 22
  on using SAM ESDSs 22
  to DTF specifications 22
REUSE attribute 10
REUSE parameter 19
  assumptions 17

SAM access of V or B records 23
SAM ESDS
  access method services for 24
  defining a 13, 14
  deleting 27
  differences between (VSAM) ESDS and 34
  extending via managed-SAM access 24
  portability restrictions 22
  restrictions on using 22
  using a 19
SAM ESDSs, supporting 8
SAM feature, how to use 13
SAM files
  converting to VSAM files 5
  DASD sharing 8
  extending existing 10
SAM logical block size 19, 36
SAM logical record size 19, 36
SAM to VSAM control 11
SAM, unsupported device dependent functions 22
SAM, valid logical record formats 23
sample programs and job streams 29
secondary allocation 9
  size 19
security and integrity of data 7
security and password protection 7
security levels 7
setting up a quantity of space 13
shared access, levels of 8
SHAREOPTIONS parameter 20
  assumptions 17
  treatment 8
simplified job control 6
single extent allocation 18
single extent primary allocation 16, 18
SIZE parameter 21
sizes, requesting 7
sources of VSAM information 17
space management
  automatic 9
  for SAM feature, how to use 13
space, setting up 13
SPEED mode, extending files in 10
SPEED parameter assumptions 17
suballocating space for SAM ESDS 13
SUBALLOCATION parameter assumptions 17
support of catalog recovery 8
support of multiple extents 8
support of multiple volumes 8
symbolic unit use 21

tracks, number of 19
TRACKS parameter in DEFINE CLUSTER command 15
TYPEFLE=OUTPUT 16
TYPEFLE=WORK 16, 21, 22
unauthorized access, preventing 7
UNORDERED parameter assumptions 17
unsupported device dependent SAM functions 22
USECLASS parameter assumptions 17
using a SAM ESDS 19
using SAM/VSAM combinations 5
using VSAM with SAM 5

V records, SAM access of 23
valid combinations of access methods and file types 12
valid SAM logical record formats 23
VB records, SAM access of 23
VERIFY command 27
volume serial number 19
VOLUMES parameter 20
  in DEFINE CLUSTER command 16
VSAM access considerations 23
VSAM access of a SAM ESDS 23
VSAM assumptions 17
VSE/VSAM space management for SAM feature, how to use 13

ways to use SAM and VSAM 6
work files 9, 16
  DTFSD specifications for 18
WRITECHECK parameter 24

Using the VSE/VSAM Space Management for SAM Feature (File No. S370-30) Printed in U.S.A. SC24-5192-0

SC24-5192-0

Using the VSE/VSAM Space
Management for SAM Feature
SC24-5192-0

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

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.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

• What is your occupation? _____

• How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

SC24-5192-0

Reader's Comment Form

Cut or Fold Along Line

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold | Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

Using the VSE/VSAM Space Management for SAM Feature (File No. S370-30) Printed in U.S.A. SC24-5192-0

Using the VSE/VSAM Space
Management for SAM Feature
SC24-5192-0

READER'S COMMENT FORM

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

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.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

	Yes	*No*
• Does the publication meet your needs?	☐	☐
• Did you find the material:		
Easy to read and understand?	☐	☐
Organized for convenient use?	☐	☐
Complete?	☐	☐
Well illustrated?	☐	☐
Written for your technical level?	☐	☐

• What is your occupation? _____

• How do you use this publication:

As an introduction to the subject?	☐	As an instructor in class?	☐
For advanced knowledge of the subject?	☐	As a student in class?	☐
To learn about operating procedures?	☐	As a reference manual?	☐

Your comments:

If you would like a reply, please supply your name and address on the reverse side of this form.

Thank you for your cooperation. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments.)

Reader's Comment Form

Cut or Fold Along Line

Using the VSE/VSAM Space Management for SAM Feature (File No. S370-30) Printed in U.S.A. SC24-5192-0

Fold and Tape | Please Do Not Staple | Fold and Tape

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

International Business Machines Corporation
Department 6R1BP
180 Kost Road
Mechanicsburg, PA 17055

Fold | Fold

If you would like a reply, *please print:*

Your Name _____

Company Name _____ *Department* _____

Street Address _____

City _____

State _____ *Zip Code* _____

IBM Branch Office serving you _____

IBM®

SC24-5192-0