SC24-5216-1
File No. S370-30

Program Product

Using the VSE/VSAM Backup/Restore Feature

IBM

SC24-5216-1
File No. S370-30

Program Product

Using the VSE/VSAM Backup/Restore Feature

Release 2

Program Number 5746-AM2

IBM

Second Edition (July 1981)

This edition, SC24-5216-1 is a major revision of SC24-5216-0. It applies to Release 2 of the Virtual Storage Extended/Virtual Storage Access Method (VSE/VSAM) Backup/Restore Feature, which is a part of Program Product 5746-AM2, and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. Changes are periodically made to the information contained 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.

Summary of Amendments

For a description of changes, see page iii.

A change to the text or to an illustration is indicated by a vertical line to the left of the change.

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 copies of 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 Corporation, 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.

Summary of Amendments for Using the VSE/VSAM Backup/Restore Feature

Summary of Amendments for SC24-5216-1 VSE/VSAM Backup/Restore Feature Release 2

VSE/VSAM Backup/Restore Release 2 lets you perform the following actions:

- Back up and restore empty objects.
- Restore objects to a DASD volume of a different device type than the backup volume. You can move objects in the following ways:
  - From one CKD device to another CKD device;
  - From one FBA device to another FBA device;
  - From a CKD device to an FBA device;
  - From an FBA device to a CKD device.
- Change the allocation size for the data component of an object at restoration (new DATARECORDS parameter).
- Change the index CI size at restoration (new INDEXCISIZE parameter).

Preface

The purpose of this manual is to provide the information necessary for the use of the VSE/VSAM Backup/Restore Feature by VSE/VSAM system and application programmers. You are presumed to have a general knowledge of VSE/VSAM concepts and job control.

This manual is organized in the following way:

- *Chapter 1: VSE/VSAM Backup/Restore* provides an overview of the facilities and functions available when the feature is installed.
- *Chapter 2: Using VSE/VSAM Backup/Restore* provides a detailed explanation of file backup and restoration procedures.
- *Chapter 3: The BACKUP and RESTORE Commands* provides a description of the BACKUP and RESTORE commands and their parameters.
- *Chapter 4: Storage and Library Considerations* provides storage estimates and system library requirements for Backup/Restore users.
- *Chapter 5: Examples* provides examples of backup and restoration job streams.

Prerequisite Publications

You 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/VSAM Programmer's Reference, SC24-5145

Related Publications

VSE/VSAM Documentation Subset, SC24-5191

VSE/VSAM Messages and Codes, SC24-5146

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

Contents

Chapter 1: VSE/VSAM Backup/Restore

The VSE/VSAM Backup/Restore Feature is a significant extension to VSE/VSAM Access Method Services. It consists of two additional Access Method Services commands: BACKUP and RESTORE. It specifically addresses the function of backing up VSAM files onto tape and restoring them.

Data is the most critical resource in data processing and should therefore be backed up regularly. This allows you to restore the latest backup copy in case the data is damaged or destroyed.

The VSE/VSAM Backup/Restore Feature provides:

- a wide spectrum of functions warranting optimum backup/restoration efficiency;
- ease of use as an integral part of the functions offered;
- inherent high performance with additional performance tuning options.

Wide Range of Functions

VSE/VSAM Backup/Restore comprises two commands: BACKUP and RESTORE. These commands are part of the Access Method Services component of VSE/VSAM.

The BACKUP command produces a backup copy of the specified VSAM objects.

From the backup copy, the RESTORE command recreates a VSAM object that is equivalent to the original one. Equivalent in this sense means that the reconstructed VSAM object may have a physical structure different from that of the original object, but that it behaves, from the user's point of view, exactly the same as the original object did.

VSAM Objects Supported

The VSE/VSAM Backup/Restore Feature can back up the following objects and their catalog information onto tape for later restoration:

- key-sequenced data sets (KSDS)
- entry-sequenced data sets (ESDS)
- relative-record data sets (RRDS)
- alternate indexes (AIX)
- SAM ESDSs in control-interval format

Empty objects can be backed up and restored. An empty object is an object that was defined using the NOALLOCATION parameter, an object that has never been loaded with data, or an object that has not been loaded since being reset.

Though they cannot be specified in the command, paths are backed up and restored automatically when their respective path entry cluster is backed up or restored.

Magnetic Tape Units Supported

VSE/VSAM Backup/Restore supports backing up onto tape and restoring from tape on all magnetic tape units supported by VSE.

The IBM 8809 Magnetic Tape Unit is supported in both start/stop and streaming modes. Although VSE/VSAM Backup/Restore is designed to assist streaming in a multiprogramming environment as much as possible, streaming is ensured only in single-batch environments with certain buffer sizes. Streaming is not ensured when file modifications (moving files to a volume of a different device type, or specifying new data component allocation size or index control interval size) are made at restoration.

Target Volume/Location for Restoration

The objects backed up can be restored to the same or to a new set of volumes. The device type of the new volumes need not be the same as that of the original volumes.

You can migrate files:

- From one type of FBA device to another type of FBA device;
- From one type of CKD device to another type of CKD device;
- From an FBA device to a CKD device;
- From a CKD device to an FBA device.

The only restriction in moving files to a different device is that the data CI size will not be changed in the transition. This is necessary to avoid record-level reorganization with its undesirable performance characteristics.

You should not indiscriminately migrate files from one device to another because during migration allocation sizes may change to accommodate the new device characteristics.

When restoring to the original volume, the location of suballocated files on the volume is mainly controlled by VSAM. For UNIQUE files, you must specify the location you wish the files to be restored to.

Backing up or Restoring Multiple VSAM Objects

VSE/VSAM Backup/Restore allows you to back up or restore multiple VSAM objects with a single command. The individual objects are specified in the form of a list. You can specify up to 255 VSAM objects with a single command.

Alternate indexes and paths are automatically backed up with their related base clusters or path entry clusters and thus do not reduce the number of objects that can be specified.

Using Generic Names

It may often be desirable to back up or restore several related VSAM objects under one common entryname. The VSE/VSAM Backup/Restore Feature permits you to specify a generic name to identify a group of related objects instead of specifying the individual entrynames for all the related objects.

Using a generic name, you can back up all VSAM objects of a catalog or a desired subset of objects. Using a generic name, you can restore all objects of a backup file or a subset of objects.

Excluding Objects from Backup/Restore

The generic name you want to use for backup or restoration may comprise some object(s) you do not wish to back up or restore. These objects can be excluded from backup or restoration by specifying them (in the form of an entryname or another generic name) in the EXCLUDE parameter of the BACKUP or RESTORE command.

The EXCLUDE parameter can also be used to suppress automatic backup/restoration of alternate indexes, path entries, and empty objects.

Reorganizing Control Areas

Restoration of a KSDS causes the reorganization of the data component on the basis of control areas. Control areas that were not physically adjacent, because of control area splits, are physically adjacent after restoration so that unnecessary disk arm movements for sequential processing are avoided. (Note that the relative byte addresses of the control areas will change accordingly.)

VSE/VSAM Backup/Restore can therefore be used to reorganize KSDSs on the basis of control areas.

Changing a File's Allocation Size at Restoration

You can change the amount of space allocated to a file at restoration by specifying new primary and secondary allocation sizes for it. This capability is particularly useful if you restore objects from a VSE/VSAM Backup/Restore Release 1 backup file onto a volume of a device type other than they were backed up from. The DATARECORDS parameter lets you override the original allocation information with values appropriate to the new device type.

Catalog Migration

You cannot actually back up and restore catalogs, but when you back up and restore objects (including empty objects), their catalog information is backed up and restored too.

This makes it possible for you to use BACKUP and RESTORE to copy objects and their catalog information into a different catalog. If the new catalog already contains an entry with the same name as the object being restored, that entry is deleted. Based on the information in the tape backup file, a new object is defined with the same name, and the file is restored.

Ease of Use

The VSE/VSAM Backup/Restore Feature provides a number of functions that significantly reduces the specification effort and make the functions easy to use.

Reduced Specification Effort

Multiple VSAM objects can be backed up or restored with a single command. Moreover, the use of generic names allows you to identify all objects of a catalog (or a subset of objects) with a single generic entryname. Additionally, undesired objects can be excluded from backup or restoration by use of the EXCLUDE parameter.

The combined use of entryname lists and generic and selective specification of objects greatly reduces the specification effort. Generic backup and restoration of empty objects means you no longer have to recreate empty objects when restoring objects into a catalog that has been destroyed.

The specification effort is even further reduced by the fact that alternate indexes and paths are automatically backed up or restored. This automatic backup or restoration can be suppressed if desired.

Tape Handling

VSE/VSAM Backup/Restore provides several functions that facilitate tape handling.

Alternate Tape Support/Sequential Mounting

During backup, processing switches automatically to the next backup volume on the alternate tape drive when the current backup volume has been filled (assuming an alternate tape has been assigned).

When alternate tapes are assigned and objects are restored sequentially, processing switches automatically to the next volume if:

- the current object is continued on the next backup volume, or
- the physically next object is to be restored and is the first object on the next backup volume.

Mounting the next volume on the free alternate tape drive ensures an uninterrupted flow of processing. This also gives the operator more time in which to mount the tapes and allows him to overlap rewinding of the previous backup volume with processing of the next backup volume.

Selective Tape Mounting

VSE/VSAM Backup/Restore allows you to restore a multi-volume backup file selectively. This means that you need only mount those backup volumes actually required for restoration, that contain the data for the objects to be restored. Which volumes should be mounted can be determined from the backup cross-reference listings printed upon completion of execution of the BACKUP command. Any volume not required for restoration can be skipped.

When objects are to be restored from backup volumes that precede the currently mounted backup volume, the RESTORE command will request mounting of the specific backup volume(s) containing the required data when the volume is actually needed.

Backup Cross-Reference Listings

The backup file may be a multi-volume tape file. Because of the multi-object backup and generic-name capabilities, it may not be apparent how the objects are spread over the individual tape volumes.

Therefore, at the end of processing of each BACKUP command, VSE/VSAM Backup/Restore prints two backup cross-reference listings of all objects backed up and their distribution on the tape volumes. The first listing is sorted by volume sequence number, whereas the second is sorted by object name.

The backup cross-reference listings identify all empty objects. You can use this information to identify empty objects in a catalog.

Error Handling

VSE/VSAM Backup/Restore differentiates between object-specific and system-related errors.

If an object-specific error is encountered, an appropriate error message is issued and the backup or restoration process continues with the next object to be backed up or restored. The process is terminated only if more than 32 object-specific errors have been encountered.

For system-related errors, the execution of the BACKUP or RESTORE command is terminated immediately.

Backup files whose processing was prematurely terminated can, however, be used for restoration of those objects that were successfully backed up.

RESTORE reports the progress of the total restoration operation by means of messages. This allows easier backout in case an error occurs.

When the restoration of an object cannot be successfully completed, RESTORE deletes the object again if it was already defined. This ensures that the catalog always reflects the correct status and contains no "dead" entries due to the failure of a restore operation.

High Performance

The VSE/VSAM Backup/Restore Feature is specifically geared to high performance, both in terms of execution speed and low processor utilization.

Inherent High Performance

VSE/VSAM Backup/Restore provides the following inherent performance advantages:

- It allows use of the 8809 Magnetic Tape Unit (typically used for backing up and restoring data) in streaming mode. This results in considerable time savings, especially when performing massive backup jobs at the end of the day.
- The high performance of VSE/VSAM Backup/Restore also applies to the 2400- and 3400-series Magnetic Tape Units, especially for tapes with high tape speeds and densities.
- Alternate tape support ensures uninterrupted processing for multi-volume backup files.
- Parts of a backup file that are not to be restored can be skipped.
- It lends itself to being used in a multiprogramming environment:
  - It operates with low processor utilization, thus enabling you to "squeeze in" a quick backup or restore job during normal operation without significantly impacting other jobs.
  - It can be loaded into the Shared Virtual Area (SVA), thus allowing concurrent execution in different partitions from a single program copy.

    In case of concurrent usage, the working set is bound to be smaller, which lowers the frequency of page faults. This, in turn, improves overall system performance.

Performance Tuning Options

VSE/VSAM Backup/Restore uses common buffers for reading and writing of both disk and tape information. The size and number of these buffers determine the performance. You can specify the size of the buffers for backup and the number of buffers for both backup and restoration, thus optimizing performance according to the parameters of your installation.

In addition, you may request the buffers to be fixed in storage via the job control UPSI command. This may yield an additional performance gain depending on the specific workload conditions in your installation.

Tuning capabilities are limited when you restore objects to a volume of a different device type, or if you make other file modifications such as changing the data component's allocation size or the index control interval size.

Fast KSDS Reorganization on Control-Area Basis

As mentioned earlier, VSE/VSAM Backup/Restore can be used to reorganize KSDSs on the basis of control areas in a very economical way.

Performance Considerations When Making File Modifications at Restoration

Specifying a different use class on the RESTORE command has no appreciable effect on RESTORE performance. When you restore a file to a volume of a different device type or change a file's data allocation information or index control interval size, however, it means that the space allocation and blocking characteristics of the restored file will be different from the space allocation and blocking characteristics of the file that was backed up. In order to process these changes, RESTORE command execution will take longer than when none of these changes is specified. For RRDSs and ESDSs (including SAM ESDSs), streaming mode on the 8809 Magnetic Tape Unit may be affected. For KSDSs, you can expect a significant performance degradation; streaming mode probably cannot be maintained. The actual degradation depends on device geometry, control area size, and the number of control interval splits and record deletions that have occurred.

High Function/Performance Trade-Offs

The concurrent availability of BACKUP/RESTORE and EXPORT/IMPORT provides significantly higher function and performance trade-offs.

Backup/Restoration Versus File Transportability

VSE/VSAM Backup/Restore operates on the basis of large block processing, thus achieving high performance and low processor utilization. BACKUP/RESTORE should, therefore, be used for regularly backing up your files and for restoring them when necessary.

Using BACKUP/RESTORE has the side benefit that control areas as a whole are reorganized automatically during this process.

EXPORT/IMPORT operates on either a control-interval or a record basis. Execution is, therefore, slower for high-speed tapes and requires a significantly higher percentage of processor utilization than BACKUP/RESTORE. Therefore, EXPORT/IMPORT should be used for:

- migration between systems;
- reorganization at a level lower than control areas.

Selectivity for Reorganization

With the availability of VSE/VSAM Backup/Restore, you can now select the degree of reorganization depending on your exact needs.

Reorganization on Control-Area Basis

Using VSE/VSAM Backup/Restore allows you to reorganize a KSDS on control-area basis in a fast and economical way.

VSE/VSAM Backup/Restore actually "dumps" the control areas of a file, including empty control intervals. That is, it processes empty control intervals. Whenever your file has a significant number of empty control intervals, it should, therefore, be reorganized via EXPORT/IMPORT to reduce unnecessary I/O operations on subsequent backups.

For the day-to-day or weekly backup procedure, you should use BACKUP and RESTORE with the benefit of reorganizing control areas as a whole, which reduces disk arm movements during sequential VSAM processing of the file.

Backup copies of your files will normally be available in your installation. This type of reorganization can, therefore, be accomplished simply by restoring the latest backup copy.

Reorganization on Control-Interval Basis

The function of reorganizing files on a control-interval basis is provided by the EXPORT CI-mode. The requirement for reorganization at this level is normally less frequent than for reorganization on control-area basis. Therefore, the higher cost in terms of execution time and processor utilization and, for high-speed tapes, of execution time associated with EXPORT/IMPORT, is also incurred less frequently.

Reorganization when Restoration is to a Volume of Different Device Type: When restoration is to a volume of a different device type, the file is reorganized on a control interval basis. Control intervals are loaded into the file in key sequence. Free CIs are inserted into the new control areas so that the file will again have the percentage of free CIs specified by the FREESPACE parameter.

Reorganization on Record Basis

This function is performed by EXPORT/IMPORT and implies a complete reorganization of the file(s). The requirement for this type of reorganization is even less frequent than for reorganization on control-interval basis. Due to the associated "cost," this function should be used only when absolutely required.

Compatibility and Coexistence

The following paragraphs discuss the coexistence of the backup files with the portable files created by EXPORT/EXPORTRA, as well as the compatibility of backup files with previous VSE/VSAM releases or OS/VS.

Compatibility with EXPORT/IMPORT Commands

The format of the backup file produced by VSE/VSAM Backup/Restore differs from the format of the portable files produced by EXPORT/EXPORTRA. (The difference is caused by performance considerations implemented in Backup/Restore.) Consequently, portable files created by EXPORT/EXPORTRA cannot be restored via RESTORE, nor can files created by BACKUP be processed by IMPORT/IMPORTRA. RESTORE checks, however, whether the tapes mounted for restoration actually belong to a backup file and generates a message in case they do not.

The incompatibility between files created by BACKUP and EXPORT should have no practical effect because the scope of application of the two groups of commands does not interfere with one another.

Because of interspersed tape marks, labeled tape files created by BACKUP cannot be processed as labeled tapefiles by DTFMT.

For the same reason, labeled tape files created by BACKUP cannot share tape volumes with other labeled tape files.

Compatibility with Release 1 of Backup/Restore

Files backed up under VSE/VSAM Backup/Restore Feature Release 1 can be restored on VSE/VSAM Backup/Restore Feature Release 2. Backup files created under VSE/VSAM Backup/Restore Feature Release 2 and not containing empty objects can be restored on VSE/VSAM Backup/Restore Feature Release 1.

Restoring a backup file created on Backup/Restore Release 1 will function correctly on Backup/Restore Release 2, but if restoration is to a different device type and the DATARECORDS allocation parameter is not specified on the RESTORE command, then the amount (approximate number of records) of space in the primary and secondary allocations will, in general, be different from the original allocations. Depending on the original space allocation specifications, the same number of tracks or cylinders will be allocated on the new device as had been allocated on the old device.

A backup file containing empty objects that were backed up on Backup/Restore Release 2 cannot be restored on Backup/Restore Release 1.

Backup/Restore Release 1 could not back up certain files created prior to VSE/VSAM Release 1. This was because VSE/VSAM would assign different physical record sizes to these files than they had been created with. Backup/Restore Release 2 removes this restriction for most files. Because of the physical record size modifications, however, performance degradation can be expected in such a restoration.

Compatibility with Previous Releases of VSE/VSAM and DOS/VS

VSAM files created by VSE/VSAM Releases 1 and 2 can be backed up and restored without restriction. Files, catalogs, Access Method Services jobstreams, and VSE/VSAM user programs created under VSE/VSAM Backup/Restore Release 2 and not containing or utilizing new external functions can be processed on prior releases of VSE/VSAM.

VSAM files created by DOS/VS releases prior to VSE/VSAM Release 1 can usually be backed up and restored, even though the file's physical record size might change.

VSE/VSAM Backup/Restore does not back up spanned ESDSs that cannot be restored. It generates a message stating that the file is not backed up for this reason. (Figure 2-8 shows which spanned ESDSs can be backed up.)

Access Method Services job streams and VSAM applications created under prior releases are not impacted by Backup/Restore.

Access Method Services job streams that contain BACKUP or RESTORE commands cannot be executed using releases of DOS/VS or VSE/VSAM Release 1.

Files created by RESTORE and using functions of VSE/VSAM Release 2 cannot be processed under prior releases.

EXPORT/EXPORTRA and IMPORT/IMPORTRA can be used concurrently with BACKUP/RESTORE, but their tape files cannot be intermixed or interchanged with tape files created by BACKUP.

Compatibility with OS/VS VSAM

Backup/Restore Release 1 did not back up certain files created by OS/VS because VSE/VSAM would assign different physical record sizes to these files than they had been created with. Backup/Restore Release 2 removes this restriction for most files. Because of the physical record size modifications, however, performance degradation can be expected in such a restoration. Figure 2-8 shows which files cannot be restored.

OS/VS VSAM objects created using the "old" catalog structure (before OS/VS Data Facility/Extended Function Release 1) can usually be backed up and restored by VSE/VSAM, even though the objects' physical record size might change.

There is only one way you can copy OS/VS VSAM objects created using the "new" catalog structure (beginning with OS/VS Data Facility/Extended Function Release 1) to VSE. While running on OS/VS DF/EF, you must use EXPORT (or EXPORTRA) to copy the OS/VS DF/EF files to tape. Then you can use IMPORT (or IMPORTRA) to copy the files from tape to VSE/VSAM.

In either case, backup files created by BACKUP cannot be restored under OS/VS. You must use EXPORT/IMPORT or EXPORTRA/IMPORTRA to copy files from VSE to OS/VS.

Chapter 2: Using VSE/VSAM Backup/Restore

This chapter describes how to use the BACKUP and RESTORE commands. It is structured such that it can be inserted into Chapter 2 of the manual *Using VSE/VSAM Commands and Macros.*

BACKUP: Creating a Backup Copy of a VSAM Object

The BACKUP command is used to write one or more VSAM objects onto tape for later restoration. An object to be backed up may be empty. An empty object is an object that was defined using the NOALLOCATION parameter, an object that has never been loaded with data, or an object that has not been loaded since being reset.

All file options that can be specified in the Access Method Services DEFINE command are supported for VSAM objects.

Magnetic Tape Considerations for Backup

The BACKUP command creates a labeled or unlabeled tape file, depending on the objects specified. Only standard (EBCDIC) labels are supported.

The backup file must always be assigned to SYS005 (as for EXPORT) and always starts at the beginning of a volume. When a volume has been filled during backup, the tape is rewound and unloaded.

The backup file is a single- or multi-volume tape file consisting of several subfiles separated by tape marks. These tape marks prevent labeled backup files created by BACKUP from being processed as labeled tape files by DTFMT. For the same reason, labeled backup files cannot share a tape volume with other labeled tape files.

Because VSE/VSAM Backup/Restore uses DTFMT for OPEN and EOV processing, there must be a tape mark at the beginning of each tape volume (as required by DTFMT) before you can use a brand-new tape volume for backup or change the tape density.

Alternate Tape Support for Backup

If you assign alternate tapes for backup, processing will automatically switch to the alternate tape drive and back as the individual tape volumes are filled (see Figure 2-1).

Backup Cross Reference Listings

Because the backup file can be a multi-volume tape file, it may not be apparent how the VSAM objects on the file are spread over the individual tape volumes of the backup file.

After processing of each BACKUP command, VSE/VSAM Backup/Restore therefore prints two backup cross reference listings which show how the individual VSAM objects are distributed over the volumes of the backup file. For this purpose, the individual volumes are consecutively numbered in their mounting sequence.

The first listing, the backup volume cross reference, is sorted by volume sequence number as shown in Figure 2-2.

The second cross reference listing, the backup object cross reference, is sorted by the entrynames of the VSAM objects on the backup file as shown in Figure 2-3.

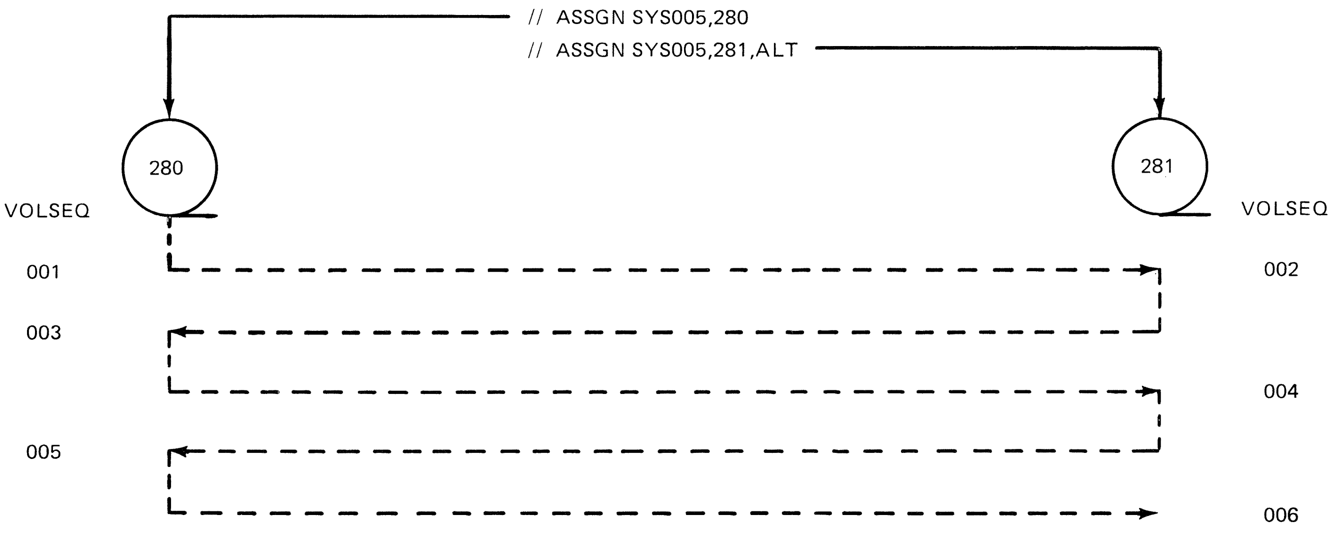


Figure 2-1. Alternate Tape Support for Backup

```
BACKUP VOLUME CROSS REFERENCE LISTING (BVCR)
VOLSEQ   VOLSER   OBJECT NAME                                   OBJECT TYPE  SEGMENT TYPE
001      TP6722   HJO.TEST.X1 ............................. KSDS         FIRST
002      TP5732   HJO.TEST.X1 ............................. KSDS         INTERMEDIATE
003      TP5834   HJO.TEST.X1 ............................. KSDS         LAST
                  HJO.TEST.X8 ............................. RRDS         ONLY
                  HJO.TEST.X4 ............................. ESDS         ONLY
                  HJO.TEST.X3 ............................. ESDS         FIRST
004      TP4910   HJO.TEST.X3 ............................. ESDS         LAST
                  DEFAULT.MODEL.ESDS ...................... ESDS         EMPTY
                  DEFAULT.MODEL.ESDS.SAM .................. SAM ESDS     EMPTY
                  SAM.WORK.FILE1 .......................... SAM ESDS     EMPTY
                  HJO.TEST.X6 ............................. KSDS         ONLY
                  HJO.TEST.X2 ............................. ESDS         ONLY
                  HJO.TEST.X7 ............................. ESDS         ONLY
                  HJO.TEST.X5 ............................. RRDS         ONLY
```

Figure 2-2. Backup Volume Cross Reference Listing

```
BACKUP OBJECT CROSS REFERENCE LISTING (BOCR)
OBJECT NAME                                 OBJECT TYPE   VOLSEQ   VOLSER   SEGMENT TYPE
DEFAULT.MODEL.ESDS ........................ ESDS          004      TP4910   EMPTY
DEFAULT.MODEL.ESDS.SAM .................... SAM ESDS      004      TP4910   EMPTY
HJO.TEST.X1 ............................... KSDS          001      TP6722   FIRST
                                                          002      TP5732   INTERMEDIATE
                                                          003      TP5834   LAST
HJO.TEST.X2 ............................... ESDS          004      TP4910   ONLY
HJO.TEST.X3 ............................... ESDS          003      TP5834   FIRST
                                                          004      TP4910   LAST
HJO.TEST.X4 ............................... ESDS          003      TP5834   ONLY
HJO.TEST.X5 ............................... RRDS          004      TP4910   ONLY
HJO.TEST.X6 ............................... KSDS          004      TP4910   ONLY
HJO.TEST.X7 ............................... ESDS          004      TP4910   ONLY
HJO.TEST.X8 ............................... RRDS          003      TP5834   ONLY
SAM.WORK.FILE1 ............................ SAM ESDS      004      TP4910   EMPTY
```

Figure 2-3. Backup Object Cross Reference Listing

Explanations

VOLSEQ is the sequence number in which the BACKUP command processed the individual tape volumes.

VOLSER is the volume serial number of the appropriate tape volume for labeled tapes. For unlabeled tapes, **NONE** is printed under this heading.

SEGMENT TYPE indicates which part of the VSAM object is contained on the referenced volume:

ONLY -
the indicated tape volume containes the complete VSAM object;

FIRST -
the indicated tape volume contains the first part of the VSAM object;

INTERMEDIATE -
the indicated tape volume contains an intermediate part of the VSAM object;

LAST -
the indicated tape volume contains the last part of the VSAM volume.

EMPTY -
the indicated object was found, but only its catalog description was placed in the backup file because no data is present.

Generic Names

If you wish to back up several related (by name) VSAM objects at a time, you can specify a *generic name* in the BACKUP command instead of the individual entrynames of all the related objects if the names have the same leading character groups (name segments). A generic name as supported by VSAM Backup/Restore is an entryname that contains an asterisk (*) as the last name segment.

A generic name represents a group of entrynames, namely the set of all entrynames contained in the default catalog (job or master catalog) that start with the characters preceding the asterisk in the generic name. Empty objects are included. Figure 2-4 shows the effect of using a generic name for backup.

There is no formal restriction on the number of VSAM objects that can be backed up with a generic name. There are, however, practical restrictions because each VSAM object requires a small amount of virtual storage.

Using a generic name provides an easy way to back up *all* VSAM objects of a catalog. To achieve this, simply specify "*" in the BACKUP command.

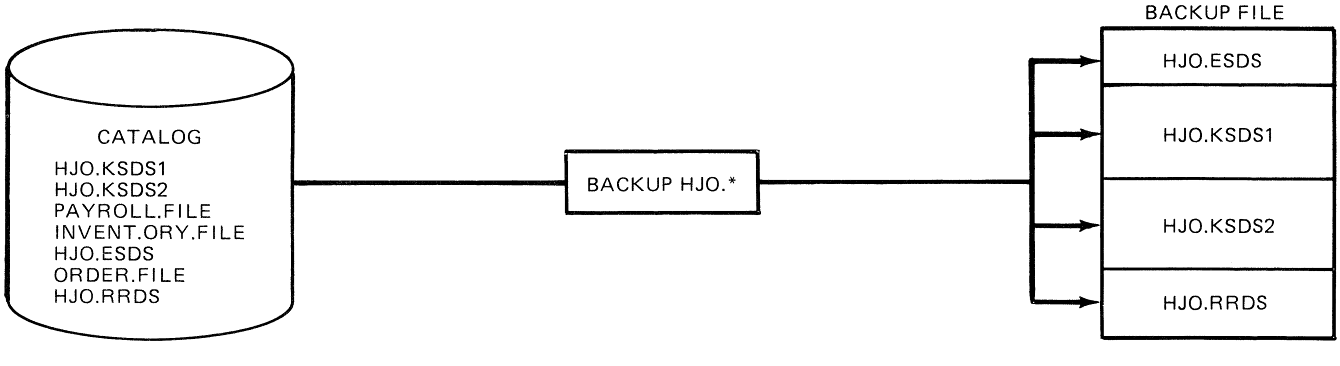


Figure 2-4. Generic Backup

Excluding Objects from Backup

If not all objects represented by a generic name are to be backed up, use the EXCLUDE parameter. EXCLUDE takes precedence over the entryname list; none of the objects named in the EXCLUDE parameter (individually or by generic name) are backed up, even if they are specified in the entryname list. This is illustrated in Figure 2-5.

Note: If no object is found for a specified entryname (whether or not the object was excluded), a warning message is printed so that you can remove the superfluous entryname the next time you run the backup or restoration job.

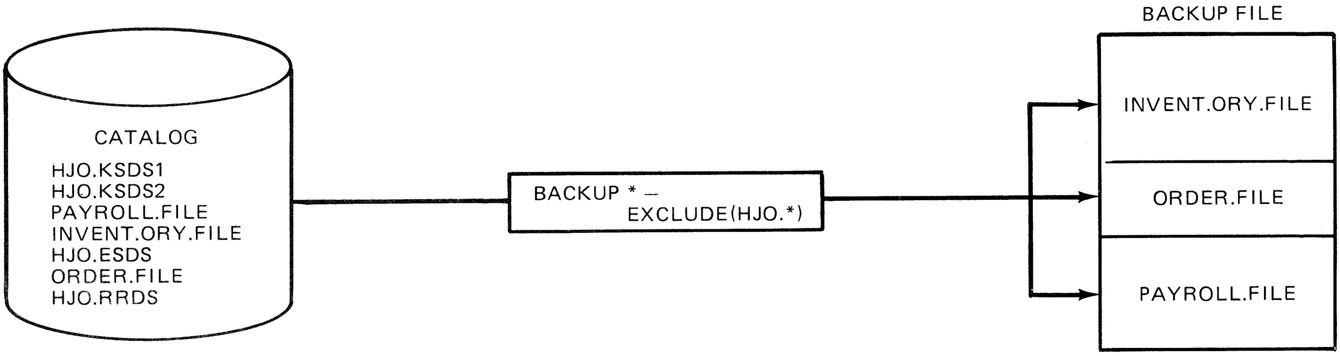


Figure 2-5. Exclusion from Backup

Excluding Associated Objects

When an object is a candidate for backup, all its associated objects are automatically backed up with it. Such automatic backup can be suppressed by explicitly naming the objects to be suppressed in the EXCLUDE parameter. Figures 2-6 and 2-7 show how this can be achieved.

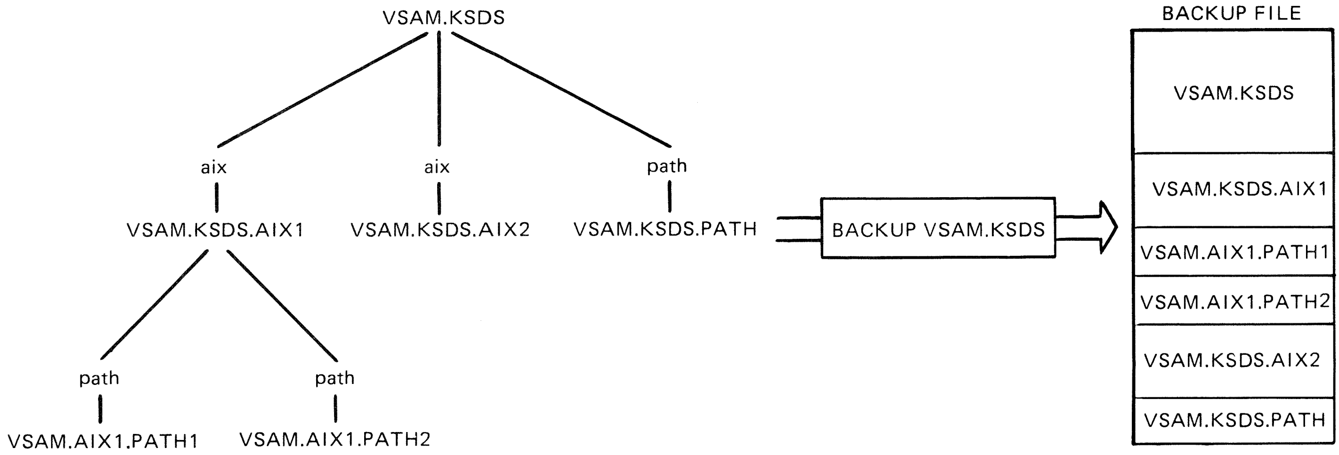


Figure 2-6. Automatic Backup of Alternate Indexes and Paths

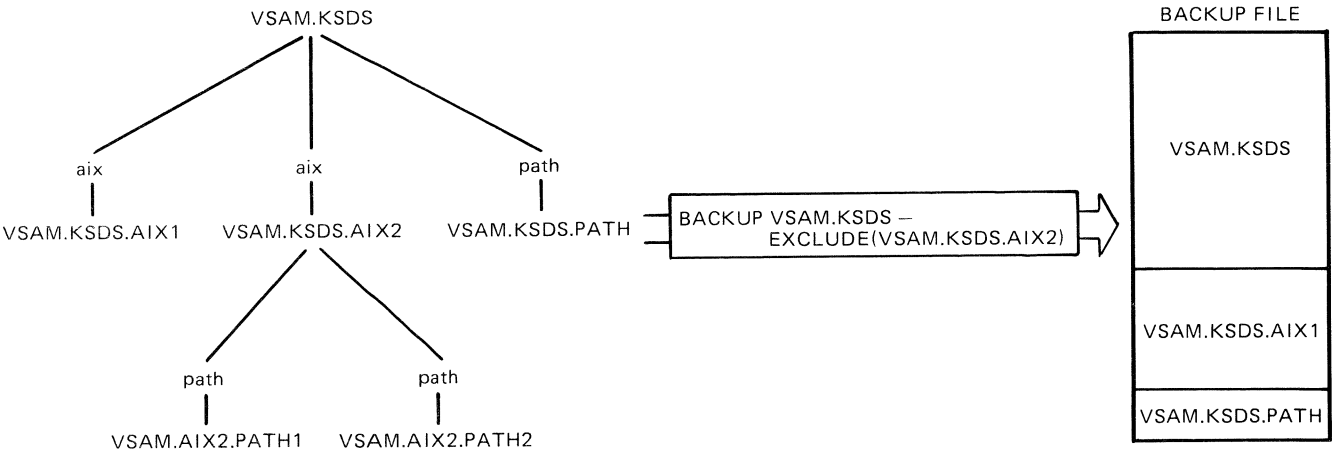


Figure 2-7. Exclusion from Automatic Backup

Buffer Allocation for Backup

VSE/VSAM Backup/Restore uses buffers for reading and writing both disk and tape information. The size and number of these buffers determine the performance.

The size of the buffers is specified in the BLOCKSIZE parameter of the BACKUP command. The size determines the size of the tape blocks of the backup file and influences the streaming of the 8809 Magnetic Tape Unit.

The number of buffers available to the backup function is specified in the BUFFERS parameter of the BACKUP command. This parameter is intended to assist streaming in a multiprogramming environment.

The trailing reflective marker on any backup volume must be sufficiently far from the end of the tape reel to accommodate as many buffers as specified in the BUFFERS parameter, plus some small control records.

Whether or not the buffers should be fixed in real storage can be specified via the UPSI job control statement.

VSE/VSAM Backup/Restore optimizes resource allocation according to the device used. Under normal circumstances, you need not specify the BLOCKSIZE and BUFFERS parameters. You may use them, however, to tune operation of your system.

Backing Up a Spanned ESDS

VSAM files created by VSE/VSAM can be backed up by means of VSE/VSAM Backup/Restore without restriction. However, spanned ESDSs that were created by OS/VS or DOS/VS VSAM cannot always be backed up and restored by means of VSE/VSAM Backup/Restore. The additional physical record size support provided by VSE/VSAM may result in a different physical record size for the data component (after redefinition of the file during restoration) from the one that existed when the file was backed up. This is particularly true for files for which the physical record size is different from the control interval size. However, a one-time use of EXPORT/IMPORT will remedy the situation.

Figure 2-8 shows where physical record size conflicts exist. Refer to the device type representing the volume from which the file is to be backed up.

Data Control Interval Size	Device Type			
	2314 2319	3330	3340 3344	3350
.5K				
1K				
1.5K	*	*	*	*
2K				
2.5K		*	*	*
3K	*	*	*	*
3.5K		*	*	*
4K				
4.5K	*	*	*	*
5K			*	*
5.5K	*	*	*	*
6K	*	*	*	*
6.5K	*		*	
7K	*		*	*
7.5K	*	*	*	*
8K			*	*
10K			*	*
12K	*	*		*
14K	*		*	*
16K			*	*
18K	*	*	*	*
20K				*
22K				*
24K	*	*	*	*
26K	*			
28K	*			*
30K	*	*	*	*
32K			*	*

* VSE/VSAM BACKUP/RESTORE cannot be used for backup of a spanned ESDS created by a release prior to VSE/VSAM Release 1 or by OS/VS.

Figure 2-8. Control Interval Sizes Supported

Job Control Statements for BACKUP

The job control statements to be used in connection with the BACKUP command are shown below. It is assumed that the DLBL statement for the master catalog has been placed into the permanent standard label area, as follows:

```
// DLBL IJSYSCT,'VSAM.MASTER.CATALOG',,VSAM
```

If the file is cataloged in a user catalog, you must provide a DLBL statement identifying that catalog in the format:

```
// DLBL IJSYSUC,'VSAM.USER.CATALOG',,VSAM
```

The UPSI statement can be used to fix the buffers used by BACKUP in real storage. A left-hand bit of 1 in the UPSI byte indicates that the buffers are to be fixed while being used. A left-hand bit of 0 (the default) indicates that fixing is not desired.

```
 // JOB     jobname
[// UPSI    1  ]                Specify only if buffers are to be fixed
 // ASSGN   SYS005,cuu          Must be SYS005, as for EXPORT
[// ASSGN   SYS005,cuu,ALT]     Alternate tape assignment if desired

 // TLBL     dname,'file-id'[,date]
 // EXEC     IDCAMS,SIZE=AUTO
    BACKUP     •
               •
               •
             [STDLABEL(dname) ]

 /*
 /&
```

File-id is the name the backup file will receive.

For the 8809 Magnetic Tape Unit, streaming is specified via the mode setting of the IPL ADD command or the ASSGN job control statement.

RESTORE: Restoring a VSAM Object from the Backup File

The RESTORE command restores the VSAM objects contained on a backup file. Restoration can be to the same volumes as those on which the object originally resided, or to different volumes. The new volumes need not be of the same device type as the original volumes. If the object is restored to the same volumes, its new location can (and generally will) be different from the original location. Restoration deletes any old copy of a VSAM object (if it is defined in the catalog effective for the RESTORE command) regardless of its retention date.

Objects to be restored are automatically defined; you need not execute a DEFINE command before restoring the objects. The automatic definition also reflects any modifications you may have specified in the RESTORE command (for example, a new set of volumes onto which you wish the objects to be restored).

RECOVERY and WRITECHECK options, if any, are recorded in the catalog information during backup, but ignored during restoration for the sake of fast file restoration.

Generic Names

As with the BACKUP command, you can specify either individual objects or a generic name, with or without the EXCLUDE parameter (see "Generic Names" in this chapter). Note that the meaning of a generic name is slightly different for RESTORE. For RESTORE, the generic name applies to the objects of the backup file, rather than to the objects of a catalog as for BACKUP. Figure 2-9 shows an example of restoring objects with a generic name.

Excluding Objects from Restoration

Excluding objects from the entryname list by means of the EXCLUDE parameter causes automatic exclusion of any associations of those objects. If, however, an association is a member of the candidate list (specified in the OBJECTS parameter), it is restored even if its base cluster is excluded from restoration (see Figure 2-10).

Restoration to a Volume of the Same Device Type

If a VSAM object is to be restored to the same volume(s) on which it originally resided, the location of the restored suballocatable files on the volume(s) will be determined by VSAM. You can, however, influence the location of such files through the USECLASS, DATAUSECLASS, and INDEXUSECLASS parameters. These

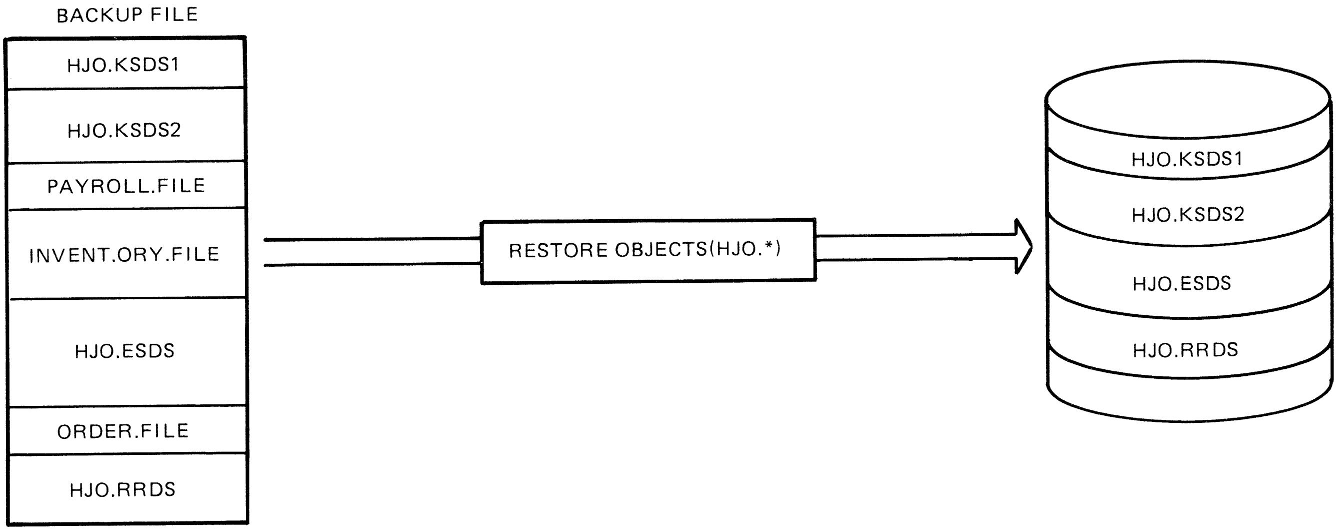


Figure 2-9. Generic Restoration

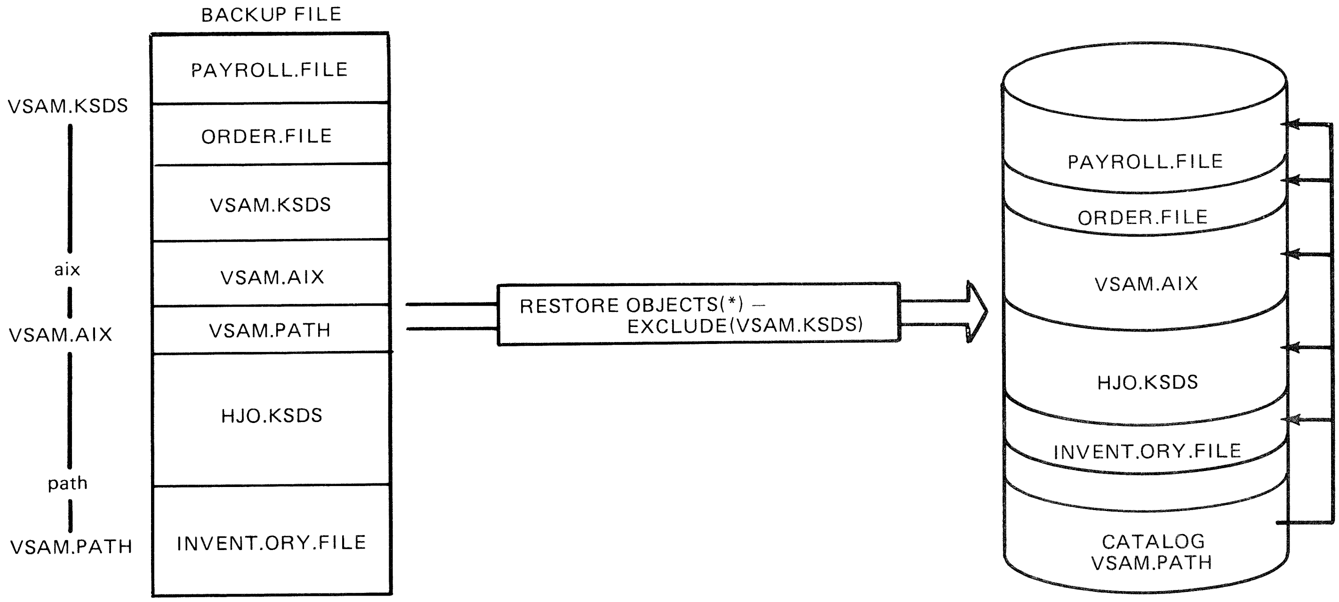


Figure 2-10. Restoration of an Alternate Index Excluding the Base Cluster

parameters allow you to select space classes (for the data and/or index component of an object) during restoration that are different from the originally chosen space classes.

An important application of these parameters is the following: Assume that the original space class represented a fixed-head area that still contains some VSAM files, but that its remaining free space (possibly the space from which the file was backed up) is damaged. Without moving the intact files contained in the fixed-head space, the data space as such cannot be deleted. Nor does VSAM mark the damaged space as defective. If you do not change the space class, VSAM tries to suballocate the space into which the file is to be restored out of the "damaged space," and the restoration will be unsuccessful. With the USECLASS, DATAUSECLASS, and INDEXUSECLASS subparameters, the file or component in question can be taken out of the fixed-head space class and moved to another, non-defective space class. For an example, see "Restoration Example 9" in Chapter 5.

For *unique* files, you must specify, in the RESTORE command, the filename of the DLBL/EXTENT statements indicating the location where you want your files to be restored. The location may be the same as the original location or it may be a different one. For an example, see "Restoration Example 3" in Chapter 5.

Restoration with File Modifications

You can make any or all of the following modifications to your files when you restore them:

- Move files to a space of a different use class;
- Move files to a volume of a different device type;
- Change the allocation size of the data component of a specific file;
- Change the size of the index control interval of a specific file.

Specifying a new use class has no appreciable effect on the performance of the RESTORE command. If you request any of the other file modifications, however, one or more of the following attributes of the cluster is likely to change:

- CA size
- Physical record size
- Index CI size
- Space allocation size.

Because of these possible modifications, you should not indiscriminately migrate files from one device to another, or change the data component allocation size or index CI size. These file modifications can result in degraded performance during RESTORE execution and changed space allocation sizes due to the new device characteristics.

Restoration to a Volume of a Different Device Type

To restore a file to a volume of different device type, simply specify *volser* in the VOLUMES subparameter (or the DATAVOLUMES or INDEXVOLUMES subparameters) to represent the new volume. You can copy your files onto volumes of a different device type without specifying VOLUMES if one of the following situations has occurred:

- *Volser* has been assigned to a volume of a different device type since the BACKUP was performed; or
- The controlling catalog (during restoration) owns a volume of a different device type that contains the same *volser.*

The only restriction in moving files to a different device is that the data CI size must not be changed in the transition. This is necessary to avoid record-level reorganization, with its undesirable performance characteristics. Spanned ESDSs cannot be restored to a volume of a different device type if a change in the data component CA size would be required.

Note that Release 2 backup stores allocation information in device-independent units. When using a Release 2 backup file, you should not have to specify the DATARECORDS parameter to adjust allocation sizes unless you were dissatisfied with the storage allocation of the original object.

Be aware that if you

- Back up a file on device type A,
- Restore it to device type B,
- Back it up from B,
- Restore it to A,

it will probably no longer fit into its original storage allocation on device A. This is because VSAM generally needs to round up the storage allocation request when moving objects from one device type to another.

Restoring Objects from a Release 1 Backup File to a Volume of a Different Device Type

If you are restoring objects from a Backup/Restore Release 1 backup file to a volume of a different device type, you may need to specify the DATARECORDS parameter. In Release 1, allocation information was stored in device-dependent units. When you restore to a volume with a different geometry, the stored allocation information may result in inappropriate allocation sizes. The DATARECORDS parameter allows you to override the allocation information stored in the backup file with allocation values more appropriate to the new device.

One particular problem you can encounter in restoring a file to a volume of a different device type involves the following situation:

- The file is on a Backup/Restore Release 1 tape.

- The storage allocation information is not appropriate to the new device. You need to specify DATARECORDS.
- The use class specification for the data component is not the same as for the index component (use class/component separation).

When you specify DATARECORDS, VSAM ignores the Release 1 device-dependent allocation units and instead allocates storage as you specified for the *data* component. Because of use class/component separation, the allocation amount used for the *index* component is the same as when the file was backed up. If this existing index allocation amount is not enough, the RESTORE will fail. Message IDC31338I will be issued. If the RESTORE fails, perform steps 1 and 2 listed below.

If the job runs to completion, the resulting index allocation, although sufficient, may be undesirable to you. The index may be spread out over a number of extents, adversely affecting performance. Run a LISTCAT and examine the index layout. Depending on the size of the file, if the index extends over more than a few extents, it would probably be worthwhile for you to restructure the index into fewer extents. If the index layout is satisfactory to you, you do not need to do anything. If you want to reallocate the index component, use the following procedure.

1. Restore the file again, specifying that data and index are to reside in the *same* use class. When you specify only one use class, VSAM can calculate an index allocation size appropriate to the volume on which the file is to be restored.
2. Back up the output of step 1, and restore it again, this time specifying *different* use classes for data and index to reestablish use class/component separation.

Using RESTORE to Add Entries to a Catalog

Backup/Restore processes both the catalog information for an object and the object itself. This means that you can add objects to a catalog simply by specifying the name of the new catalog at restoration. VSAM will copy the catalog information for the objects and the objects themselves. For empty objects, only the catalog information will be copied. If the new catalog already contains an entry with the same name as the object being restored, that entry is deleted. Based on the information in the tape backup file, a new object is defined with the same name, and the file is restored.

Be aware that an object may be defined in more than one catalog at once. If you intend to move (not just copy) an object from one catalog to another catalog, you must delete the object from the source catalog after using BACKUP and RESTORE to copy the object's information into the new catalog.

You can merge objects defined in several different catalogs into one catalog by restoring the objects in the order you want them to be defined in the new catalog.

Subsetting During Restoration

The RESTORE command allows subparameters (such as the VOLUMES, DATAVOLUMES, or the USECLASS subparameters) for the individual entrynames. For a generic name, the specified subparameters apply to all objects of the generic name.

It is possible to exempt objects represented by a generic name from the subparameters. This can be achieved by specifying these objects with their respective subparameters in addition to the generic name. The subparameters of the generic name then apply to all objects represented by the generic name except those explicitly named with other subparameters.

For example, if the volume serial number ABCVOL is specified for the entryname ABC.* and VOLXYZ is specified for ABC.KSDS.*, then the object ABC.KSDS.#1 is restored onto the volume VOLXYZ whereas the object ABC.ESDS is restored onto ABCVOL. For an example, see "Restoration Example 2" in Chapter 5.

This technique is called subsetting and applies to all subparameters, including passwords for an entryname. If multiple entrynames form subsets of one another, the subparameters applicable to an object are those specified for the entryname that is closest ("best fit") to the object's name.

Subparameters (except passwords) never apply to objects that are automatically restored. If you want to change the volumes or the useclass of such objects, you must name them in the OBJECTS parameter together with the subparameters you wish to apply.

Reorganization of Control Areas

The restoration of a KSDS causes a reorganization of the data component on the basis of control areas. The individual records of a control interval and the control intervals of a control area are not reorganized, but the control areas as a whole are. Control areas that previously were not physically adjacent because of control area splits will be physically adjacent after restoration, thus avoiding unnecessary disk arm movements for sequential processing.

The reorganization of KSDS control areas as a whole results in a change of the relative byte addresses.

Magnetic Tape Considerations for Restoration

For restoration, the backup file must always be assigned to SYS004 as for IMPORT.

The interspersed tape marks of the backup file allow individual objects that are not to be restored to be skipped during restoration. In this way the tape channel can be freed for the duration of the skipping operation.

When all specified files have been restored from a backup volume, the tape is unloaded.

Selective Tape Mounting for Restoration

The restoration function does not require that the first volume of the backup file be mounted when the execution of the RESTORE command begins.

Each volume of the backup file contains a directory by means of which Backup/Restore determines which of the VSAM objects to be restored are on the mounted (or later) volumes of the backup file and which are on earlier volumes.

VSE/VSAM Backup/Restore restores the specified VSAM objects in the order in which they are stored on the backup file. It starts at the beginning of the mounted volume with the first object that is not an association, proceeds to the end of the backup file, and then continues with the first volume of the backup file (wrap-around). The initially mounted volume has to be mounted again if it contains part of an object that starts on an earlier volume, or if it starts with an associated object.

When the operator is asked to mount a later volume, he may skip any volumes of the backup file that do not contain objects to be restored. The cross reference listing printed upon completion of the BACKUP command will help him to determine the volumes to be mounted.

For VSAM objects that start on earlier volumes than the initially mounted one, VSE/VSAM Backup/Restore knows the exact volume and asks the operator for this volume. For the correct tape volume to be mounted, a message is printed specifying the volume sequence number assigned by the BACKUP command and listed in the cross reference listing. Restoration continues after the operator has mounted the correct tape volume.

You do not lose the ability to skip unwanted objects by not mounting the first tape volume of the backup file.

All volumes of the backup file contain the backup file creation time. Whenever a new volume is mounted, it is ensured that the new and the old volumes have the same backup file creation timestamp. Otherwise, an error message is issued that asks the operator for the correct volume. Processing can proceed after the correct volume has been mounted.

Sequential Tape Mounting for Restoration

A VSAM object to be restored may span several tape volumes of the backup file. When processing of a volume has been completed, the volume is rewound and unloaded, and processing continues with the next part of the object.

Each tape volume of the backup file terminates with an end-of-tape (EOT) record containing a volume termination timestamp that is identical with the volume creation timestamp of the next tape volume. The backup *volume* creation time stamp is in addition to the backup *file* creation time stamp. When an object spans multiple tape volumes and a transition from one tape volume to the next takes place, the termination timestamp of the volume just completed and the volume creation timestamp of the next volume are compared. If they do not match, an error message is issued that requests mounting of the correct volume. Processing can proceed after the correct volume has been mounted.

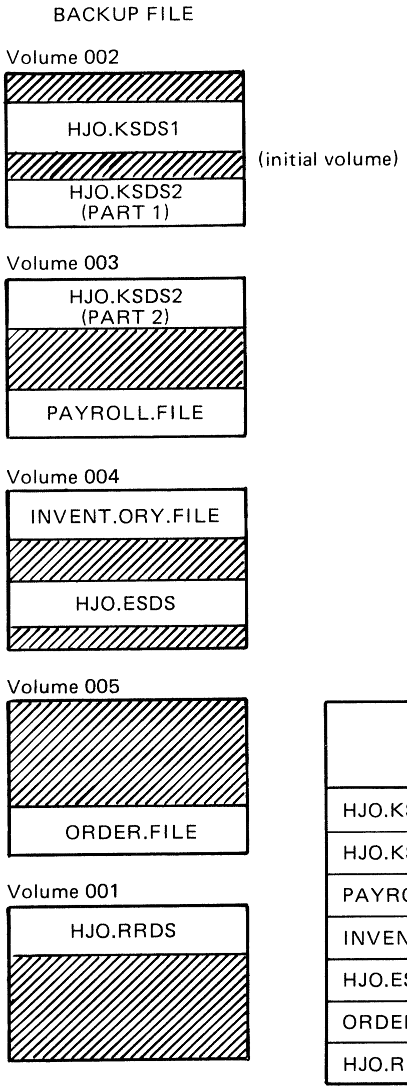
Alternate Tape Support for Restoration

If you assign alternate tapes for restoration, processing will automatically switch to the alternate tape drive and back as the individual volumes are processed (see Figure 2-11).

Buffer Allocation for Restoration

With restoration, VSE/VSAM Backup/Restore automatically chooses a buffer size equal to the appropriate buffer size for backup. No user specification is required.

The BUFFERS parameter can, however, be used to influence the number of buffers available for restoration. This parameter is intended to assist streaming of the 8809 Magnetic Tape Unit in a multiprogramming environment. The maximum number of buffers used for restoration is that used for backup. If a larger number is specified, it is reduced to the permitted maximum.



```
// JOB      RESTORE
// ASSGN    SYS004,280
// ASSGN    SYS004,281,ALT
// EXEC     IDCAMS,SIZE=AUTO
   RESTORE  OBJECTS ((HJO.*) –
                      (PAYROLL.FILE) –
                      (INVENT.ORY.FILE) –
                      (ORDER.FILE))
/*
/&
```

Object	Primary Drive 280	Alternate Drive 281	Message	Remarks
HJO.KSDS1	002			
HJO.KSDS2	002	003		Object spans volumes
PAYROLL.FILE		003		
INVENT.ORY.FILE	004			Next sequential object
HJO.ESDS	004			
ORDER.FILE	005		4191I/4192A	Not next object
HJO.RRDS	001		4190A	Not next object, 001 earlier volume

Figure 2-11. Alternate Tape Support for Restoration

Job Control Statements for Restoration

The job control statements to be used in connection with the RESTORE command are shown below. It is assumed that the DLBL statement for the master catalog has been placed into the permanent standard label area with the following job control statement:

```
// DLBL IJSYSCT,'VSAM.MASTER.CATALOG',,VSAM
```

The UPSI statement can be used to fix the buffers used by Backup/Restore. A left-hand bit of 1 on the UPSI byte indicates that the buffers are to be fixed in real storage while being used. A left-hand bit of 0 (the default) indicates that fixing is not desired.

```
 // JOB      jobname
[//UPSI     1   ]           Specify only if buffers are to be fixed
 // ASSGN   SYS004,cuu      Must be SYS004 as for IMPORT
[// ASSGN   SYS004,cuu,ALT]  Alternate tape assignment if desired

 // TLBL    dname,'file-id'

 // DLBL    dname-i3,,,VSAM
 // EXTENT  ,volser,,,rel-begin,size

 // DLBL    dname-i2,,,VSAM
 // EXTENT  ,volser,,,rel-begin,size
      •
      •
 // EXEC    IDCAMS,SIZE=AUTO
    RESTORE OBJECTS(...(entryname-i
                                 •
                                 •
                         DATAFILE(dname-i2)
                         INDEXFILE(dname-i3)
                                •
                                •
                             )•••
                          )
            STDLABEL(dname)
                 •
                 •
 /*
 /&
```

Just as for IMPORT, the backup file must be assigned to SYS004.

file-id: (in the TLBL statement) must be the name of the backup file if it has standard (EBCDIC) tape labels.

rel-begin: is the relative track number (for CKD devices) or the relative block number (for FBA devices) of the starting location of the extent for the *unique* file that is being restored.

The TLBL and DLBL/EXTENT statements are, of course, only required if the corresponding parameters are specified in the RESTORE command. For the 8809 Magnetic Tape Unit, streaming is specified via the mode setting of the IPL ADD command or the ASSGN job control statement.

Chapter 3: The BACKUP and RESTORE Commands

This chapter details the specification for the BACKUP and RESTORE commands. The format of each command is shown, followed by a discussion of each parameter.

The structure of this chapter is such that the description of each command can be inserted into Chapter 3 of the manual *Using VSE/VSAM Commands and Macros*, at the appropriate places.

BACKUP

The BACKUP command (abbreviated form: BUP) is used to produce a backup copy of one or more of the following VSAM objects:

- Key-sequenced file (KSDS)
- Entry-sequenced file (ESDS)
- Relative-record file (RRDS)
- Alternate index (AIX)
- SAM entry-sequenced file in control-interval format

The object to be backed up may be empty. (An empty object is one that was defined using the NOALLOCATION parameter, has never been loaded with data, or has not been loaded since being reset.) All file options that can be specified in the Access Method Services DEFINE command are supported for VSAM objects. The RECOVERY and WRITECHECK options are recorded in the catalog information, but ignored during restoration for the sake of fast file restoration.

When an object is backed up, its associations (alternate indexes and paths) are automatically backed up with it unless explicitly excluded.

The format of the BACKUP command is:

BACKUP	(*entryname* [/*password*] . . .) [BLOCKSIZE(*size*)] [BUFFERS(*number*)] [EXCLUDE(*entryname* . . .)] [STDLABEL(*dname*)]

For more information and examples see Chapters 2 and 5.

BACKUP Parameters

(*entryname* [/*password*] . . .)

names the object(s) to be backed up.

entryname - is the name of an object or set of objects that are candidates for backup. It can be either the full object name or a generic name. A generic name is an entryname that contains an asterisk (*) as the last name segment. It represents all entrynames (including empty objects) contained in the default (job or master) catalog that start with the characters preceding the * in the generic name.

Note that a generic name is an abbreviation for a set of *explicitly* named objects.

If only one entryname is specified, the parentheses surrounding the entryname/password may be omitted.

If you want to back up all objects of the catalog, specify "*" for entryname.

You can specify up to 255 entrynames with a single command. This number is not reduced by any alternate indexes and paths, which are automatically backed up with their related base clusters or alternate indexes.

password - specifies the password of the object to be backed up if it is password protected. The password specified must be the file or catalog master password if the protection information of the object is to be backed up with the object for reinstallment during restoration. If the associated entryname is a generic name, all objects represented by the generic name must have the same master password unless the master password of the default job or master catalog is specified.

If the password specified is not the master password, the object(s) will be backed up without password and a message is issued to indicate this fact. The minimum password you must specify is the file's read password. Otherwise the object(s) will not be backed up. If the read password for an object indicated by a generic name is specified, all objects represented by the generic name must have the same read password.

BLOCKSIZE(*size*)

specifies the approximate size (in bytes) of the blocks (buffers) to be written onto the backup file. For each object to be backed up, the specified buffer size is rounded to accommodate an integral number of physical blocks of the object's data component. For key-sequenced files, the buffer size is made at least as large as the index control-interval size. If the backup device is an 8809 Magnetic Tape Unit, the minimum buffer size required for streaming is enforced.

Do not confuse the terms *blocksize* and *physical record size.* Blocksize refers to data on *tape,* while physical record size pertains to data stored on *disk.*

If you do not specify a BLOCKSIZE value, VSE/VSAM Backup/Restore determines the buffer size based on the DASD device characteristics, the physical characteristics of the file, and the minimum buffer size requirements for streaming mode on the IBM 8809, if this is the backup device.

The blocksize must not be smaller than 512 bytes. It must not be larger than 32K bytes for S/370 mode or 64K - 1 byte for ECPS: VSE mode.

Streaming considerations for the 8809: Because the buffer size determines the size of the tape blocks, it also determines the number of tape I/O operations that are needed for backing up or restoring an object. Increasing the buffer size also increases the time available to VSE/VSAM Backup/Restore for re-instructions to the tape device, which enables you to positively influence the system's streaming capability under different workload conditions. The larger the specified BLOCKSIZE value, the more likely that streaming is achieved and the less processor time is required per transferred byte, thus increasing the multiprogramming capability.

Abbreviation: BLKSZ

BUFFERS(*number*)

specifies the number of common data buffers to be used for the backup operation. (The size of each buffer is derived from the BLOCKSIZE specification.) By means of this parameter, you can vary the performance behavior of the backup operation in a multiprogramming environment and influence the streaming ability of the backup function. Increasing the number of buffers increases the streaming potential.

The minimum number of buffers that can be specified is 2, the default is 3, and the maximum is 255.

Note that backup will write the number of buffers specified in the BUFFERS parameter following the trailing reflective marker of a tape volume. Therefore, the marker must be far enough from the end of the tape volume to allow space for the buffers.

Abbreviation: BFRS

EXCLUDE(*entryname* . . .)

specifies a list of VSAM objects that are not to be backed up even though they are named (for example, via a generic name) in the list of VSAM objects for backup.

Entryname - may be the name of a single object or a generic name. Up to 255 entrynames can be specified. You may specify the names of empty objects to be excluded.

If an object is excluded from backup, any alternate indexes and paths defined for the object are also excluded, unless they (the alternate indexes) are explicitly named in the list of backup objects.

Note that a generic name is an abbreviation for a set of *explicitly* named objects. That is to say, all objects comprised by the generic name are interpreted as being explicitly named. In particular, an entry name of "*" explicitly names an alternate index.

If an object is backed up, but an alternate index defined for it is excluded from backup, the alternate index (and any paths defined for it) are automatically deleted when the object is restored.

Abbreviation: EXCL

STDLABEL(*dname*)

is required if the backup file is to contain standard (EBCDIC) tape labels, in order to prevent accidental destruction of other labeled tape files or of the backup file itself.

dname - specifies the *filename* of the TLBL statement that contains the necessary label information.

Abbreviation: SLBL

RESTORE

The RESTORE command (abbreviated form: RST) is used to reinstall the objects of a backup file that was produced via the BACKUP command. All objects or individual objects of the backup file can be restored. Generic restoration is also possible.

All alternate indexes contained on the backup file are automatically restored with their related base cluster. Also, all paths are restored when their associated alternate index or path entry cluster is restored.

The format of the RESTORE command is:

```
RESTORE   OBJECTS (  (entryname [/password ]
                                 [DATAFILE(dname )]
                                 [DATARECORDS(primary [ƀsecondary ])]
                                 [DATAUSECLASS(primary [ƀsecondary ])]
                                 [DATAVOLUMES(volser ...)]
                                 [FILE(dname )]
                                 [INDEXCISIZE(size )]
                                 [INDEXFILE(dname )]
                                 [INDEXUSECLASS(primary [ƀsecondary ])]
                                 [INDEXVOLUMES(volser ...)]
                                 [USECLASS(primary [ƀsecondary ])]
                                 [VOLUMES(volser ...)]
                     )...
                  )
            [BUFFERS(number )]
            [CATALOG(catname [/password ])]
            [EXCLUDE(entryname ...)]
            [STDLABEL(dname )]
            [USECLASS(primary [ƀsecondary ])]
            [VOLUMES(volser ...)]
```

For more information and examples see chapters 2 and 5.

RESTORE Parameters: Summary

File Modification at Restoration: The following parameters are required only if you want to change the characteristics of the specified object when it is redefined in the catalog during restoration:

VOLUMES
DATAVOLUMES
INDEXVOLUMES
USECLASS
DATAUSECLASS
INDEXUSECLASS
DATARECORDS
INDEXCISIZE

By means of these parameters, either the whole object or its data or index component can be moved onto different volumes or to different space classes.

Performance Degradation: If you specify that a file is to reside on a volume of a different device type, or if you specify DATARECORDS or INDEXCISIZE, VSAM usually reorganizes the file's control areas or control intervals. You will experience degraded performance compared to running RESTORE without such modifications.

Changing a file or component's use class has no appreciable effect on performance.

Global/Local Specification: Note that theVOLUMES and the USECLASS parameters can appear on two levels: they can be effective only for the entryname for which they are specified (local specification), or they can be effective for *all* specified entrynames unless specifically overridden (global specification). For example:

```
RESTORE OBJECTS((A VOLUMES(VOL001))
                         .
                         .
                         .
                        )
             VOLUMES(VOL002)
```

In this example, all objects will be restored onto VOL002, *except* object A which will be restored onto VOL001.

Unique Files: The following parameters are only applicable to a *unique* file:

FILE
DATAFILE
INDEXFILE

These parameters are used to identify the DLBL/EXTENT statements that contain the space information for a unique file or its unique components. If no unique files are to be restored, you can ignore these parameters.

If you do not have to specify any of the parameters listed above, the RESTORE command is reduced to the following format:

```
RESTORE      OBJECTS((entryname [/password ])...)
             [BUFFERS(number )]
             [CATALOG(catname [/password ])]
             [EXCLUDE(entryname ...)]
             [STDLABEL(dname )]
```

The following table shows which keywords can be used with the RESTORE command. The "Usage" column specifies whether the parameter can apply to the object as a whole (O), to the data component (D), or to the index component (I).

RESTORE Parameter	Abbr.	Usage	Example	Notes
Required parameters:				
OBJECTS	OBJ		OBJ((...))	
DATAFILE	DFILE	D	DFILE(DAT)	1
FILE	—	ODI	FILE(FILA)	1
INDEXFILE	IXFILE	I	IXFILE(IND)	1
Optional parameters:				
BUFFERS	BFRS		BFRS(2)	
CATALOG	CAT		CAT(UCT/PWD)	
DATARECORDS	DREC	D	DREC(500 100)	
DATAUSECLASS	DUSCL	D	DUSCL(7 P)	2
DATAVOLUMES	DVOL	D	DVOL(DVOL)	2
EXCLUDE	EXCL	O	EXCL(FLE.A)	
INDEXCISIZE	IXCISZ	I	IXCISZ(10 24)	
INDEXUSECLASS	IXUSCL	I	IXUSCL(7 P)	2
INDEXVOLUMES	IXVOL	I	IXVOL(IVOL)	2
STDLABEL	SLBL		SLBL(TAPE)	
USECLASS	USCL	ODI	USCL(7 P)	2
VOLUMES	VOL	ODI	VOL(VOL1)	2

Notes:

1. Required only if a unique file is to be restored.
2. Required only if the object or its data or index component is to be moved to a different space class or volume.

RESTORE Parameters

OBJECTS((*entryname* [/*password*]) . . .)

names the object(s) to be restored. If only one entryname is specified, the parentheses surrounding the entryname and its subparameters can be omitted.

entryname - is the name of an object or set of objects to be restored. It can be either the name of a single object or a generic name. A generic name is an entryname that contains an asterisk as the last name segment. It represents all entrynames contained on the backup file that start with the characters preceding the * in the generic name.

Note that a generic name is an abbreviation for a set of *explicitly* named objects.

If you want to restore all objects of the backup file, specify * for *entryname.*

Up to 255 entrynames can be specified with a single command.

password - specifies the password of a password-protected object contained in the catalog that is to be replaced by an object of the same name contained on the backup file. The password must be either the master password of the object to be replaced or the catalog's master password.

Thus, an object that was backed up without its passwords (because only the read password was specified) can only replace the old copy if the master password of the password-protected old copy or the catalog's master password is specified.

If a generic name is used, either the master password of all objects to be replaced must be the same, or the master password of the receiving catalog must be specified.

Abbreviation: OBJ

BUFFERS(*number*)

specifies the number of common data buffers to be used for restoration. The size of the buffers is the same as that used for the associated backup operation. The number of buffers specified should not exceed the number of buffers specified in the corresponding BACKUP command; if it does, the value is automatically reduced to that value.

If BUFFERS is not specified, the number of buffers used for the corresponding backup operation will be taken as the default.

Abbreviation: BFRS

CATALOG(*catname* [/*password*])

specifies the name and the password of the catalog in which the restored objects are to be defined. You must specify this parameter if the catalog that is intended to receive the VSAM objects to be restored is not the master catalog or the job catalog. This parameter must also be specified if the receiving catalog is password protected, or you will be prompted for the password. The password must either be the update password or a higher-level password of the catalog.

Abbreviation: CAT

DATAFILE(*dname*)

is effective only for a file with a UNIQUE data component.

dname - identifies the DLBL/EXTENT statements that contain the space information for the UNIQUE data component.

DATAFILE or FILE must be specified for a UNIQUE data component. If both are specified, DATAFILE takes effect for the data component and FILE is ignored as far as the data component is concerned.

DATAFILE can only be specified for individual VSAM objects and not on the global level.

Abbreviation: DFILE

DATARECORDS (*primary* [ƀ*secondary*])

specifies the amount of space, in units of records, to be allocated for the data component.

DATARECORDS is used to change the storage allocation for an object. This function is of particular value to you if you have files that were backed up on Backup/Restore Release 1 that you now wish to restore to a volume of a different device type. DATARECORDS can override the device-dependent units stored on the Release 1 backup tape and allocate space quantities appropriate to the new device type.

Specify the amount in decimal (n), hexadecimal (X'n'), or binary (B'n'). VSAM multiplies the values specified for primary and secondary space by the average record size of the file to determine the allocation amount for the data component. If the file is to contain free space, specify an amount large enough to accommodate the free space as well.

Specifying DATARECORDS overrides both primary and secondary allocation amounts for the data component. If you specify only the primary amount, the secondary amount defaults to zero.

For indexed files whose data and index components are to reside in space of the *same* use class (specified either at restoration or when the file was originally defined), VSAM calculates the amount of space to be allocated for the index component. If data and index are to reside in space of *different* use classes, the space allocated for the index component will be the same as when the object was backed up (that is, index space allocation will be unaffected by the DATARECORDS parameter).

You may specify this parameter for only specific entrynames (local specification). It may not be specified globally to apply to all objects. If you do not specify this parameter, the allocation size(s) saved during backup will be used to define the object for restoration.

Be aware that specifying DATARECORDS usually causes VSAM to use a new control area size for the data component. When this file modification occurs, you will experience degraded performance compared to RESTORE without file modification.

Abbreviation: DREC

DATAUSECLASS (*primary* [ƀ*secondary*])

can be specified only for individual objects. (It cannot be specified globally to apply to all objects.) It is used to modify the space classes for primary and secondary allocation of the data component of the restored object as follows:

Primary allocation of space:

0 The object is to occupy class-0 data space. Data spaces defined under OS/VS and DOS/VS Release 34 (and prior releases) are class-0 data spaces. 0 is the default space classification.

1-7 The object is to occupy any user-defined space class between 1 and 7.

Secondary allocation of space:

0 Class-0 data space is to be used by the object.

P The primary space classification is to be used. P is the default.

The space classes for the index component are not modified. The specification of DATAUSECLASS overrides any global specification or an appropriate local (pertaining to a particular entryname) specification of the USECLASS parameter as far as the data component is concerned.

Abbreviation: DUSCL

DATAVOLUMES (*volser* . . .)

can be specified only for individual objects. (It cannot be specified globally to apply to all objects.) It is used to move the data component of the restored object to a set of volumes that is different from the volume(s) from which the data component was backed up. The volumes need not be of the same device type as the backup volumes.

The volumes of the index component are not affected by this parameter.

The specification of DATAVOLUMES overrides any global specification or an appropriate local (pertaining to a particular entryname) specification of the VOLUMES parameter as far as the data component is concerned.

volser . . . denotes the list of volume serial numbers of the new volume set for the data component.

Abbreviation: DVOL

EXCLUDE (*entryname* . . .)

specifies a list of VSAM objects that are not to be restored even though a generic name in the OBJECTS parameter may include these VSAM objects.

Entryname - is the name of a single object or a generic name. Up to 255 entrynames can be specified.

If an object is excluded from restoration, any alternate indexes and paths defined for the object are also excluded, unless they (the alternate indexes) are explicitly named in the OBJECTS parameter.

Note that a generic entry in the OBJECTS parameter is interpreted as an abbreviation for the set of entrynames expressed by the generic name. All objects represented by a generic name are considered explicitly specified so that an alternate index represented by a generic name is restored even if its base cluster is excluded from restoration.

If an object is restored but an alternate index defined for it is excluded from restoration, the alternate index (and any paths associated with it) are automatically deleted.

Abbreviation: EXCL

FILE (*dname*)

can be specified only for individual VSAM objects. (It cannot be specified globally to apply to all objects.)

dname - identifies the DLBL/EXTENT statements that contain the space information for all UNIQUE components of a file to be restored.

If both components of the file are unique and reside on the same volume, you must specify at least two of the three parameters FILE, DATAFILE, or INDEXFILE, identifying different DLBL/EXTENT statements for the data and index components. FILE then applies to the components for which no explicit DATAFILE or INDEXFILE parameter is specified.

Abbreviation: none

INDEXCISIZE(*size*)

specifies, in number of bytes, the index CI size to be used in definition and restoration of an individual object. Specify the value in decimal (n), hexadecimal (X'n'), or binary (B'n'). The size can be any multiple of 512, up to 8192 bytes. If an improper multiple is coded, VSAM selects the next higher multiple. INDEXCISIZE may be specified for only specific entrynames (local specification). It cannot be specified globally to apply to all objects.

If you do not specify this parameter, the index CI size saved during backup is used to define the object for restoration. If you specify this parameter for a file that is not a KSDS, it is ignored.

You will probably not have to specify this parameter often, if at all. Typically you would use it if you have restored a file to a different device type and discovered (from LISTCAT output) that the restored file has increased unreasonably in size. (Some increase can reasonably be expected because restoration to a different device type reestablishes the CA free space percentage.) An unreasonable increase is caused by an excessive number of CIs remaining unused because entries for them could not be fit into the index records.

Be aware that specifying INDEXCISIZE forces VSAM to do a control interval reorganization. This file modification will cause degraded performance compared to RESTORE without file modification.

Abbreviation: IXCISZ, IXCNVSZ, or INDEXCNVSIZE

INDEXFILE(*dname*)

is effective only for a file that has a UNIQUE index component.

dname - identifies the DLBL/EXTENT statements that contain the space information for the UNIQUE index component.

INDEXFILE or FILE must be specified for a UNIQUE index component. If both are specified, INDEXFILE takes effect for the index component and FILE is ignored as far as the index component is concerned.

INDEXFILE can only be specified for individual VSAM objects. (It cannot be specified globally to apply to all objects.)

Abbreviation: IXFILE

INDEXUSECLASS(*primary* [ƀ*secondary*])

can be specified only for individual objects. (It cannot be specified globally to apply to all objects.) It is used to modify the space classes for primary and secondary allocation of the index component of a key-sequenced file or an alternate index.

The space classes for the data component are not modified.

The specification of INDEXUSECLASS overrides any global specification or an appropriate local (pertaining to a particular entryname) specification of the USECLASS parameter as far as the index component is concerned.

Primary and secondary have a similar meaning as described under the DATAUSECLASS parameter.

Abbreviation: IXUSCL

INDEXVOLUMES(*volser* . . .)

can be specified only for individual objects. (It cannot be specified globally to apply to all objects.) It is used to move the index component of a key-sequenced file or an alternate index to a set of volumes that is different from the volume(s) from which the index component was backed up. The volumes need not be of the same device type as the backup volumes.

The volumes of the data component are not affected by this parameter.

The specification of INDEXVOLUMES overrides any global specification or an appropriate local (pertaining to a particular entryname) specification of the VOLUMES parameter as far as the index component is concerned.

volser . . . denotes the list of volume serial numbers of the new volume set for the index component.

Abbreviation: IXVOL

STDLABEL(*dname*)

is required if the backup file contains standard (EBCDIC) tape labels. In this case, *dname* specifies the *filename* of the TLBL statement that contains the necessary label information.

Abbreviation: SLBL

USECLASS(*primary* [ƀ*secondary*])

can be specified for either a particular entryname or all entrynames. If specified on a global level, it is effective for all specified objects for which an individual USECLASS is not also specified. When specified for a particular entryname, the USECLASS parameter is effective only for the entry it is associated with and overrides any global specification.

With this parameter, you can modify the space classes for primary and secondary allocation of *all* components of the explicitly named restored objects. Primary and secondary have the same meaning as described under the DATAUSECLASS parameter.

Abbreviation: USCL

VOLUMES(*volser* . . .)

can be specified for either a particular entryname or all entrynames. If specified on a global level, it is effective for all specified objects, except:

- automatically restored objects
- for objects whose global specification is overridden by a specification for the particular entryname.

When specified for a particular entryname, the VOLUMES parameter applies only to the entry it is associated with and overrides any global specification.

With this parameter, you can move *all* components of the explicitly named restored objects to a set of volumes different from the one from which the objects were backed up. The volumes need not be of the same device type as the backup volumes. The VOLUMES parameter need not be specified if the backup copy is to be restored to the same *volser* from which the backup was made.

volser . . . denotes the list of volume serial numbers of the new volume set.

Abbreviation: VOL

Chapter 4: Storage and Library Considerations

This chapter describes storage requirements for VSE/VSAM Backup/Restore. It is structured such that it can be attached to Chapter 3 of the manual *VSE/VSAM Programmer's Reference.*

Loading VSE/VSAM Backup/Restore into the SVA

A sufficient number of entries in the System Directory List (SDL) and sufficient space in the SVA must be provided at IPL. VSE/VSAM Backup/Restore requires:

> 6 entries in the SDL and
>
> 60K bytes of storage in the SVA.

If you have no other user phases to be loaded into the SVA, the SVA command in your IPL procedure should be:

```
SVA SDL=6,PSIZE=60K
```

Otherwise, you should adjust the SVA command to your requirements.

VSE/VSAM Backup/Restore may be loaded into the SVA using the following procedure:

```
// JOB VSE/VSAM BACKUP/RESTORE SVA LOAD PROCEDURE
// EXEC MAINT
   CATALP IDCPLBP0
   SET SDL
   IDCBP01, SVA
   IDCBP03, SVA
   IDCCDBP, SVA
   IDCTSBP0, SVA
   IDCRT01, SVA
   IDCCDRT, SVA
/*
/+
/&
```

If you use ASI (automated system initialization), simply store the following statements in a job control procedure for the background partition.

```
SET SDL
   IDCBP01, SVA
   IDCBP03, SVA
   IDCCDBP, SVA
   IDCTSBP0, SVA
   IDCRT01, SVA
   IDCCDRT, SVA
/*
```

Access Method Services

When VSE/VSAM Backup/Restore is installed, the core image and relocatable libraries require the following phases and modules. These library sizes, names, and link-edit statements are in addition to those already required for Access Method Services and documented in *VSE/VSAM Programmer's Reference.*

Core Image Library

Phases	7
Blocks	72

Phases:

IDCBP01
IDCBP02
IDCBP03
IDCCDBP
IDCCDRT
IDCRT01
IDCTSBP0

Relocatable Library

Number of Modules	Number of Blocks
70	520

Modules:

IDCBPADE
IDCBPALE
IDCBPBBF
IDCBPBDR
IDCBPBDS
IDCBPBLE
IDCBPBPC
IDCBPBPO
IDCBPBPV
IDCBPCAU
IDCBPCMA
IDCBPCRB
IDCBPDDR
IDCBPDDW
IDCBPDYB
IDCBPDYS
IDCBPFSR
IDCBPLVO
IDCBPMDE
IDCBPMSH
IDCBPNBV
IDCBPOON
IDCBPOVC
IDCBPPXL
IDCBPRSL
IDCBPRVB
IDCBPRVD
IDCBPRVL
IDCBPSLE
IDCBPSRD
IDCBPSXL
IDCBPVCL
IDCBPVOP
IDCBPWOH
IDCRTACA
IDCRTBBR
IDCRTBDX
IDCRTBFV
IDCRTBRL
IDCRTCLX
IDCRTDFO
IDCRTDVO
IDCRTDWR
IDCRTDWW
IDCRTFSR
IDCRTGEX
IDCRTGNX
IDCRTMDS
IDCRTMSS
IDCRTMTL
IDCRTMTN
IDCRTMTS
IDCRTOPI
IDCRTPFO
IDCRTRDS
IDCRTRDX
IDCRTREV
IDCRTROH
IDCRTRTC
IDCRTRTO
IDCRTRVX
IDCRTWRS
IDCRTWRX

Link-Edit Statements:

```
// OPTION CATAL
   INCLUDE IDCCMZ3
// EXEC LNKEDT
// OPTION CATAL
   INCLUDE IDCCMZ4
// EXEC LNKEDT
// OPTION CATAL
   INCLUDE IDCCMZ5
// EXEC LNKEDT
// OPTION CATAL
   INCLUDE IDCCMZ6
// EXEC LNKEDT
// OPTION CATAL
   INCLUDE IDCCMZ7
// EXEC LNKEDT
// OPTION CATAL
   INCLUDE IDCCMZ8
// EXEC LNKEDT
// OPTION CATAL
   INCLUDE IDCCMZ9
// EXEC LNKEDT
```

Virtual Storage

In addition to basic VSAM and Access Method Services virtual storage requirements (described in *VSE/VSAM Programmer's Reference*), you must provide sufficient virtual storage in the partition to accommodate the BACKUP or RESTORE command.

Command	Bytes
BACKUP	43,000
RESTORE	53,000

Chapter 5: Examples

This chapter provides examples of both backup and restoration. The material is organized such that it can be easily incorporated in the appropriate VSE/VSAM documentation.

Backup Examples

For the subsequent examples, it is assumed that the following catalogs and objects exist:

Master Catalog

Name: VSAM.MASTER.CATALOG

Objects:

KSDS	VMC.KSDS.#1	
	VMC.KSDS.#2	
ESDS	VMC.ESDS.#1	
AIX	VMC.AIX.#1	based on VMC.KSDS.#1
	VMC.AIX.#2	based on VMC.KSDS.#1
Path	VMC.PATH.#1	based on VMC.AIX.#1
	VMC.PATH.BCL	based on VMC.KSDS.#1

User Catalog

Name: VSAM.USER.CATALOG.A

Objects:

KSDS	PAYROLL.FILE.BRANCH01	
	PAYROLL.FILE.BRANCH02	
	•	
	•	
	PAYROLL.FILE.BRANCH25	
	PAYROLL.CONTROL.FILE1	
	PAYROLL.CONTROL.FILE2	
	VSAM.DATA.SET.CLUSTER	
	VSAM.DATA.SET.BCL	
	CALC.KSDS	
ESDS	CALC.ESDS	
	VSAM.DATA.FILE	
	PAYROLL.SUMMARY	
AIX	AIX.CALC.KSDS	based on CALC.KSDS
	VSAM.DATA.SET.AIX1	based on VSAM.DATA.SET.BCL
	VSAM.DATA.SET.AIX2	based on VSAM.DATA.SET.BCL
	VSAM.DATA.SET.AIX3	based on VSAM.DATA.SET.BCL
Path	VSAM.DATA.SET.P1	based on VSAM.DATA.SET.AIX1
	VSAM.DATA.SET.P2	based on VSAM.DATA.SET.AIX2
	PATH.CALC.AIX	based on AIX.CALC.KSDS

It is further assumed that the following DLBL statement has been placed into the permanent standard label area:

```
// DLBL IJSYSCT,'VSAM.MASTER.CATALOG',,VSAM
```

Backup Example 1

Objectives:

- Backup of a single VSAM object from the master catalog.
- Automatic backup of the alternate indexes and paths based on the VSAM object.
- Automatic backup of the paths based on the alternate indexes of the VSAM object.
- Default buffer specification.

Specifications:

Catalog	VSAM master catalog
Entrynames	VMC.KSDS.#1
Buffers	Default number of buffers (3) Default buffer size Not fixed in real storage
Tape labels	None
Excluded objects	None

Job Stream:

```
// JOB          BACKUP1 BACKUP ONE OBJECT BY NAME
// ASSGN        SYS005,191
// EXEC         IDCAMS,SIZE=AUTO
   BACKUP       VMC.KSDS.#1
/*
/&
```

Objects Backed Up:

VMC.KSDS.#1	specified VSAM object
VMC.AIX.#1	alternate index for VMC.KSDS.#1
VMC.PATH.#1	path for VMC.AIX.#1
VMC.AIX.#2	alternate index for VMC.KSDS.#1
VMC.PATH.BCL	path based on VMC.KSDS.#1

Backup Example 2

Objectives:

- Non-generic backup of multiple VSAM objects from the job catalog.
- Specification of buffer parameters and options.
- Labeled backup file.

Specifications:

Catalog	VSAM job catalog (VSAM.USER.CATALOG.A)
Entrynames/password	CALC.KSDS CALC.ESDS PAYROLL.CONTROL.FILE1/MPWD1 PAYROLL.FILE.BRANCH01/MPWD2
Buffers	Number of buffers: 4 Buffersize: 8192 Fixed in real storage
Tape labels	BACKUP.FILE
Excluded objects	None

Job Stream:

```
// JOB    BACKUP2 BACKUP MULTIPLE OBJECTS BY NAME
// UPSI   1
// ASSGN  SYS005,191
// TLBL   TAPE,'BACKUP.FILE'
// DLBL   IJSYSUC,'VSAM.USER.CATALOG.A',,VSAM
// EXEC   IDCAMS,SIZE=AUTO
   BACKUP (CALC.KSDS CALC.ESDS -
           PAYROLL.CONTROL.FILE1/MPWD1 -
           PAYROLL.FILE.BRANCH01/MPWD2 -
          )
          BUFFERS(4) -
          BLOCKSIZE(8192) -
          STDLABEL(TAPE)
/*
/&
```

Objects Backed Up:

CALC.KSDS	specified VSAM object
AIX.CALC.KSDS	alternate index for CALC.KSDS
PATH.CALC.AIX	path for AIX.CALC.KSDS
CALC.ESDS	specified VSAM object
PAYROLL.CONTROL.FILE1	specified VSAM object
PAYROLL.FILE.BRANCH01	specified VSAM object

Note: Instead of the two entry names CALC.KSDS and CALC.ESDS, you could simplify the specification by using the generic name CALC.*.

Backup Example 3

Objectives:

- Generic backup of objects from the job catalog.
- No objects excluded.

Specifications:

Catalog	VSAM job catalog (VSAM.USER.CATALOG.A)
Entrynames	CALC.* VSAM.DATA.SET.*
Buffers	Default number of buffers (3) Default buffer size Not fixed in real storage
Tape labels	None
Excluded objects	None

Job Stream:

```
// JOB    BACKUP3 BACKUP BY GENERIC NAME
// ASSGN  SYS005,191
// DLBL   IJSYSUC,'VSAM.USER.CATALOG.A',,VSAM
// EXEC   IDCAMS,SIZE=AUTO
   BACKUP (CALC.* VSAM.DATA.SET.*)
/*
/&
```

Objects Backed Up:

CALC.ESDS	specified via CALC.*
CALC.KSDS	specified via CALC.*
AIX.CALC.KSDS	alternate index for CALC.KSDS
PATH.CALC.AIX	path for AIX.CALC.KSDS
VSAM.DATA.SET.BCL	specified via VSAM.DATA.SET.*
VSAM.DATA.SET.AIX1	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.P1	path for VSAM.DATA.SET.AIX1
VSAM.DATA.SET.AIX2	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.P2	path for VSAM.DATA.SET.AIX2
VSAM.DATA.SET.AIX3	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.CLUSTER	specified via VSAM.DATA.SET.*

Backup Example 4

Objective:

Backup of all VSAM objects of the job catalog, with the exception of a generic group of objects and one other object.

Specifications:

Catalog	VSAM job catalog (VSAM.USER.CATALOG.A)
Entryname	*
Buffers	Default number of buffers Default buffer size Not fixed in real storage
Tape labels	None
Excluded objects	PAYROLL.FILE.BRANCH01 • • PAYROLL.FILE.BRANCH25 CALC.KSDS

Job Stream:

Permanent assignment of SYS005 is assumed for this example.

```
// JOB    BACKUP4 ALL OBJECTS OF CATALOG WITH GENERIC EXCLUSION
// DLBL   IJSYSUC,'VSAM.USER.CATALOG.A',,VSAM
// EXEC   IDCAMS,SIZE=AUTO
   BACKUP (*/CATMPWD) -
          EXCL (PAYROLL.FILE.* CALC.KSDS)
/*
/&
```

Objects Backed Up:

AIX.CALC.KSDS	specified via *
PATH.CALC.AIX	path for AIX.CALC.KSDS
CALC.ESDS	specified via *
PAYROLL.CONTROL.FILE1	specified via *
PAYROLL.CONTROL.FILE2	specified via *
PAYROLL.SUMMARY	specified via *
VSAM.DATA.FILE	specified via *
VSAM.DATA.SET.BCL	specified via *
VSAM.DATA.SET.AIX1	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.P1	path for VSAM.DATA.SET.AIX1
VSAM.DATA.SET.AIX2	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.P2	path for VSAM.DATA.SET.AIX2
VSAM.DATA.SET.AIX3	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.CLUSTER	specified via *

Note: AIX.CALC.KSDS and PATH.CALC.AIX are backed up despite the fact that CALC.KSDS is exccluded from backup. This is because AIX.CALC.KSDS is "explicitly" specified by the generic name "*" causing its backup. As a result, PATH.CALC.AIX is also backed up.

If they are to be excluded from backup, AIX.CALC.KSDS must also be specified in the EXCLUDE parameter. PATH.CALC.AIX need not be specified in the EXCLUDE parameter because it is not a VSAM object and thus is only backed up if the alternate index on which it is based is backed up.

Backup Example 5

Objectives:

- Generic backup from the job catalog.
- Exclusion of one object from backup.

Specifications:

Catalog	VSAM job catalog (VSAM.USER.CATALOG.A), DLBL statement in permanent standard label area
Entryname	VSAM.DATA.*
Buffers	Default number of buffers Default buffer size Not fixed in real storage
Tape labels	None
Excluded objects	VSAM.DATA.SET.CLUSTER

Job Stream:

```
// JOB      BACKUP5
// ASSGN    SYS005,191
// EXEC     IDCAMS,SIZE=AUTO
   BACKUP   (VSAM.DATA.*) -
            EXCLUDE (VSAM.DATA.SET.CLUSTER)
/*
/&
```

Objects Backed Up:

VSAM.DATA.FILE	specified via VSAM.DATA.*
VSAM.DATA.SET.BCL	specified via VSAM.DATA.*
VSAM.DATA.SET.AIX1	alternate index of VSAM.DATA.SET.BCL
VSAM.DATA.SET.P1	path for VSAM.DATA.SET.AIX1
VSAM.DATA.SET.AIX2	alternate index of VSAM.DATA.SET.BCL
VSAM.DATA.SET.P2	path for VSAM.DATA.SET.AIX2
VSAM.DATA.SET.AIX3	alternate index of VSAM.DATA.SET.BCL

Backup Example 6

Objective:

Backup all VSAM objects of the master catalog.

Specifications:

Catalog	VSAM master catalog
Entryname	*
Buffers	Default number of buffers Default buffer size Not fixed in real storage
Tape labels	None
Excluded objects	None

Job Stream:

Permanent assignment of SYS005 is assumed for this example.

```
// JOB      BACKUP6
// EXEC     IDCAMS,SIZE=AUTO
   BACKUP   (*)
/*
/&
```

Objects Backed Up:

VMC.ESDS.#1	specified via *
VMC.KSDS.#1	specified via *
VMC.AIX.#1	alternate index of VMC.KSDS.#1
VMC.PATH.#1	path for VMC.AIX.#1
VMC.AIX.#2	alternate index of VMC.KSDS.#1
VMC.PATH.BCL	path for VMC.KSDS.#1
VMC.KSDS.#2	specified via *

Restoration Examples

The following restoration examples are based on the backup files created by the examples in the preceding section. Reference to the backup file is made by the backup job name (BACKUP3) used in the backup examples. The meaning of the objects is the same as defined at the beginning of this chapter.

It is assumed that the DLBL statement for the VSAM master catalog is contained in the permanent standard label area.

Restoration Example 1

Objectives:

- Non-generic restoration of multiple VSAM objects into the job catalog.
- Specification of the buffer parameters and options.
- Labeled backup file.
- No define modifications.
- All VSAM objects suballocatable.

Specifications:

Catalog	VSAM job catalog (VSAM.USER.CATALOG.A)
Backup file	Created by BACKUP2
Entrynames/password	AIX.CALC.KSDS PAYROLL.CONTROL.FILE1/MPWD1
Buffers	Number of buffers: 2 Not fixed in real storage
Tape labels	BACKUP.FILE
Excluded objects	None
Modifications	None

Job Stream:

```
// JOB     RESTORE1
// UPSI    1
// ASSGN   SYS004,191
// TLBL    TAPE,'BACKUP.FILE'
// DLBL    IJSYSUC,'VSAM.USER.CATALOG.A',,VSAM
// EXEC    IDCAMS,SIZE=AUTO
   RESTORE OBJECTS((AIX.CALC.KSDS) -
                   (PAYROLL.CONTROL.FILE1/MPWD1)) -
           BUFFERS(2) -
           STDLABEL(TAPE)
/*
/&
```

Restored Objects:

AIX.CALC.KSDS	specified VSAM object
PATH.CALC.AIX	path for AIX.CALC.KSDS
PAYROLL.CONTROL.FILE1	specified VSAM object

Restoration Example 2

Objectives:

- Restoration of all VSAM objects of a backup file.
- Restoration of a generic group of objects to a new set of volumes, with the exception of two objects of the generic group that are restored to different volumes.
- All VSAM objects suballocatable.

Specifications:

Catalog	VSAM job catalog, no passwords
	DLBL statement in permanent standard label area
Backup file	Created by BACKUP4
Entrynames	*with no objects excluded
	VSAM.DATA.*
	VSAM.DATA.SET.BCL
	VSAM.DATA.SET.CLUSTER
Buffers	Default number of buffers
	Not fixed in real storage
Modifications	Data components of VSAM.DATA.SET.BCL and VSAM.DATA.SET.CLUSTER to be restored onto volume DATVOL
	Index components of VSAM.DATA.SET.BCL and VSAM.DATA.SET.CLUSTER to be restored onto volume IXVOL
	Data component of all other objects of VSAM.DATA.* to be restored onto VOL001
	Index component of all other objects of VSAM.DATA.* to be restored onto VOL002
Job control	Permanent assignment for SYS004

Job Stream:

```
// JOB       RESTORE2
// EXEC      IDCAMS,SIZE=AUTO
   RESTORE OBJECTS((*) -
                   (VSAM.DATA.* -
                         DATAVOLUMES(VOL001) -
                         INDEXVOLUMES(VOL002) -
                   )
                   (VSAM.DATA.SET.BCL -
                         DATAVOLUMES(DATVOL) -
                         INDEXVOLUMES(IXVOL) -
                   )
                   (VSAM.DATA.SET.CLUSTER -
                         DATAVOLUMES(DATVOL) -
                         INDEXVOLUMES(IXVOL)
                   )
                   )
/*
/&
```

Restored Objects:

AIX.CALC.KSDS	onto original volumes
PATH.CALC.AIX	
CALC.ESDS	onto original volumes
PAYROLL.CONTROL.FILE1	onto original volumes
PAYROLL.CONTROL.FILE2	onto original volumes
PAYROLL.SUMMARY	onto original volumes
VSAM.DATA.FILE	data component onto VOL001
VSAM.DATA.SET.BCL	data component onto DATVOL index component onto IXVOL
VSAM.DATA.SET.AIX1	data component onto VOL001 index component onto VOL002
VSAM.DATA.SET.P1	
VSAM.DATA.SET.AIX2	data component onto VOL001 index component onto VOL002
VSAM.DATA.SET.P2	
VSAM.DATA.SET.AIX3	data component onto VOL001 index component onto VOL002
VSAM.DATA.SET.CLUSTER	data component onto DATVOL index component onto IXVOL

Restoration Example 3

Objective:

Restoration of VSAM objects with a UNIQUE data component

Specifications:

Catalog	VSAM master catalog, no passwords
Backup file	Created by BACKUP3
Entrynames	CALC.KSDS
Buffers	Default buffer number Default buffer size Not fixed in real storage
Excluded objects	None
Modifications	None

Job Stream:

```
// JOB     RESTORE3
// ASSGN   SYS004,191
// DLBL    DATA,,,VSAM
// EXTENT  ,VOL111,,,190,85
// EXEC    IDCAMS,SIZE=AUTO
   RESTORE OBJECTS (CALC.KSDS -
                    DATAFILE(DATA))
/*
/&
```

Restored Objects:

CALC.KSDS	specified object
AIX.CALC.KSDS	alternate index for CALC.KSDS
PATH.CALC.AIX	path for AIX.CALC.KSDS

Restoration Example 4

Objectives:

- Restoration of a single VSAM object of the backup file into the master catalog.
- Automatic restoration of the alternate indexes and paths based on that object.
- Automatic restoration of the paths based on the alternate indexes of the VSAM object.
- Default buffer specifications.
- All VSAM objects suballocatable.

Specifications:

Catalog	VSAM master catalog
Backup file	Created by BACKUP6
Entrynames	VMC.KSDS.#1
Buffers	Default number of buffers Not fixed in real storage
Tape labels	None
Excluded objects	None
Modifications	None

Job Stream:

```
// JOB      RESTORE4
// ASSGN    SYS004,191
// EXEC     IDCAMS,SIZE=AUTO
   RESTORE  OBJECTS(VMC.KSDS.#1)
/*
/&
```

Restored Objects:

VMC.KSDS.#1	specified VSAM object
VMC.AIX.#1	alternate index of VMC.KSDS.#1
VMC.PATH.#1	path for VMC.AIX.#1
VMC.AIX.#2	alternate index of VMC.KSDS.#1
VMC.PATH.BCL	path for VMC.KSDS.#1

Restoration Example 5

Objectives:

- Generic restoration of multiple VSAM objects into a specified VSAM catalog.
- No objects excluded.
- All VSAM objects suballocatable.

Specifications:

Catalog	Explicitly specified catalog (VSAM.CATALOG)
Backup file	Created by BACKUP3
Entrynames	VSAM.* CALC.*
Buffers	Default number of buffers Not fixed in real storage
Tape labels	None
Excluded objects	None
Modifications	None

Job Stream:

```
// JOB      RESTORE5
// ASSGN    SYS004,191
// EXEC     IDCAMS,SIZE=AUTO
   RESTORE  OBJECTS((VSAM.*)(CALC.*)) -
            CATALOG(VSAM.CATALOG)
/*
/&
```

Restored Objects:

CALC.ESDS	specified via CALC.*
CALC.KSDS	specified via CALC.*
AIX.CALC.KSDS	alternate index of CALC.KSDS
PATH.CALC.AIX	path for AIX.CALC.KSDS
VSAM.DATA.SET.BCL	specified via VSAM.*
VSAM.DATA.SET.AIX1	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.P1	path for VSAM.DATA.SET.AIX1
VSAM.DATA.SET.AIX2	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.P2	path for VSAM.DATA.SET.AIX2
VSAM.DATA.SET.AIX3	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.CLUSTER	specified via VSAM.*

Note: Because the two generic names VSAM.* and CALC.* make up all objects of the backup file, you could simplify the specification by using the generic name "*".

Restoration Example 6

Objectives:

- Generic restoration into the job catalog.
- Exclusion of one VSAM object from restoration.
- No define modifications.
- All VSAM objects suballocatable.

Specifications:

Catalog	Job catalog (VSAM.USER.CATALOG.A), DLBL statement in permanent standard label area, No passwords
Backup file	Created by BACKUP5
Entrynames	VSAM.DATA.SET.*
Buffers	Default number of buffers Not fixed in real storage
Tape labels	None
Excluded objects	VSAM.DATA.SET.AIX1
Modifications	None
Job Control	Permanent assignment for SYS004

Job Steam:

```
// JOB      RESTORE6
// EXEC     IDCAMS,SIZE=AUTO
   RESTORE  OBJECTS(VSAM.DATA.SET.*) -
            EXCLUDE(VSAM.DATA.SET.AIX1)
/*
/&
```

Restored Objects:

VSAM.DATA.SET.BCL	specified via VSAM.DATA.SET.*
VSAM.DATA.SET.AIX2	alternate index of VSAM.DATA.SET.BCL
VSAM.DATA.SET.P2	path for VSAM.DATA.SET.AIX2
VSAM.DATA.SET.AIX3	alternate index of VSAM.DATA.SET.BCL

Note: VSAM.DATA.SET.P1 is not restored because it is based upon VSAM.DATA.SET.AIX1 and is not a VSAM object. By the redefinition of VSAM.DATA.SET.BCL, existing versions of VSAM.DATA.SET.AIX1 and VSAM.DATA.SET.P1 are deleted.

Restoration Example 7

Objectives:

- Restoration of all VSAM objects of a backup file into the master catalog.
- No define modifications.
- All VSAM objects suballocatable.

Specifications:

Catalog	VSAM master catalog, no passwords
Backup file	Created by BACKUP6
Entrynames	*
Buffers	Default number of buffers Not fixed in real storage
Tape labels	None
Excluded objects	None
Modifications	None
Job Control	Permanent assignment for SYS004

Job Stream:

```
//  JOB      RESTORE7
//  EXEC     IDCAMS,SIZE=AUTO
    RESTORE  OBJECTS(*)
/*
/&
```

Restored Objects:

All objects of backup file.

Restoration Example 8

Objectives:

- Restoration of the VSAM objects of a backup file with the exception of a generic group of VSAM objects and one individual VSAM object.
- Restoration into the password protected master catalog.
- No define modifications.
- All VSAM objects suballocatable.

Specifications:

Catalog	VSAM master catalog (VSAM.MASTER.CATALOG), update password = UPDPW
Backup file	Created by BACKUP4
Entrynames	*
Buffers	Default number of buffers Not fixed in real storage
Tape labels	None
Excluded objects	PAYROLL.CONTROL.FILE1 PAYROLL.CONTROL.FILE2 PAYROLL.SUMMARY AIX.CALC.KSDS
Modifications	None
Job Control	Permanent assignment for SYS004

Job Stream:

```
// JOB        RESTORE8
// EXEC       IDCAMS,SIZE=AUTO
   RESTORE    OBJECTS(*) -
              EXCLUDE(PAYROLL.* AIX.CALC.KSDS) -
              CATALOG(VSAM.MASTER.CATALOG/UPDPW)
/*
/&
```

Restored Objects:

CALC.ESDS	specified via *
VSAM.DATA.FILE	specified via *
VSAM.DATA.SET.BCL	specified via *
VSAM.DATA.SET.AIX1	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.P1	path for VSAM.DATA.SET.AIX1
VSAM.DATA.SET.AIX2	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.P2	path for VSAM.DATA.SET.AIX2
VSAM.DATA.SET.AIX3	alternate index for VSAM.DATA.SET.BCL
VSAM.DATA.SET.CLUSTER	specified via *

Restoration Example 9

Objectives:

- Restoration of all VSAM objects of a backup file.
- Restoration of all objects to a common set of volumes that is different from the originating set of volumes.
- Restoration of all objects to a new space class.
- All VSAM objects suballocatable.

Specifications:

Catalog	VSAM job catalog, no passwords, DLBL statement in permanent standard label area
Backup file	Created by BACKUP6
Entrynames	*
Buffers	Default number of buffers Not fixed in real storage
Tape labels	None
Excluded objects	None
Modifications	Global change to volumes VOL001 and VOL002 Global change to space class 7, primary and secondary
Job control	Permanent assignment for SYS004

Job Stream:

```
//  JOB      RESTORE9
//  EXEC     IDCAMS,SIZE=AUTO
    RESTORE  OBJECTS(*) -
             VOLUMES(VOL001 VOL002) -
             USECLASS(7 P)
/*
/&
```

Restored Objects:

All objects of backup file.

Restoration Example 10

Objectives:

- Restoration of objects from a tape that was backed up using VSE/VSAM Backup/Restore Release 1.
- Restoration to a device of a different device type than the objects were backed up from.

Specifications:

Catalog	VSAM master catalog
Backup file	Created by BACKUP6
Entrynames	*
Buffers	Default number of buffers Not fixed in real storage
Tape labels	None
Excluded objects	None
Modifications	Move objects to a volume of a different device type
Job control	Permanent assignment for SYS004

Job Stream:

```
//  JOB      RESTOR10
//  EXEC     IDCAMS,SIZE=AUTO
    RESTORE  OBJECTS((*) -
             VOLUMES(337501))
/*
/&
```

Restored Objects:

All objects of backup file.

Restoration Example 11

Objectives:

- Generic restoration of VSAM objects from tape backup file.
- Restoration to a volume of a different device type than the objects were backed up from.

Specifications:

Volume backed up from	333001
Volume restored to	331001
Catalog	VSAM master catalog
Backup file	Created by BACKUP6
Entrynames	VMC.KSDS.*
Buffers	Default number of buffers Not fixed in real storage
Tape labels	None
Excluded objects	None
Modifications	Generic restoration to a volume of a different device type

Job Stream:

```
// JOB      RESTOR11
// ASSGN    SYS004,191
// EXEC     IDCAMS,SIZE=AUTO
   RESTORE  OBJECTS((VMC.KSDS.*-
            VOLUMES(331001)))
/*
/&
```

Restored Objects:

VMC.KSDS.#1	specified via VMC.KSDS.*, restored to 331001
VMC.AIX.#1	alternate index of VMC.KSDS.#1, restored to 333001
VMC.AIX.#2	alternate index of VMC.KSDS.#1, restored to 333001
VMC.PATH.#1	path for VMC.AIX.#1, restored to 333001
VMC.PATH.BCL	path for VMC.KSDS.#1, restored to 333001
VMC.KSDS.#2	specified via VMC.KSDS.*, restored to 331001

Note: To restore the AIXes and paths to the new device type, specify VMC.AIX.* and VMC.PATH.* as *entrynames*.

Restoration Example 12

Objectives:

- Restoration of a specific KSDS from the job catalog.
- Increasing the storage allocation for the KSDS's data component.

Specifications:

Catalog	Job catalog (VSAM.USER.CATALOG.A), no passwords
Backup file	Created by BACKUP3
Entrynames	VSAM.DATA.SET.BCL
Buffers	Default number of buffers Not fixed in real storage
Tape labels	None
Excluded objects	None
Modifications	Increase the storage allocation for the data component

Job Stream:

```
//  JOB       RESTOR12
//  ASSGN     SYS004,191
//  DLBL      IJSYSUC,'VSAM.USER.CATALOG.A',,VSAM
//  EXEC      IDCAMS,SIZE=AUTO
    RESTORE   OBJECTS((VSAM.DATA.SET.BCL-
              DATARECORDS(400 50)))
/*
/&
```

Restored Objects:

VSAM.DATA.SET.BCL	The data component of VSAM.DATA.SET.BCL is restored to an area with a primary allocation of 400 records and a secondary allocation of 50 records
VSAM.DATA.SET.AIX1	AIX based on VSAM.DATA.SET.BCL, restored to whatever its original storage allocation was
VSAM.DATA.SET.AIX2	AIX based on VSAM.DATA.SET.BCL, restored to whatever its original storage allocation was
VSAM.DATA.SET.AIX3	AIX based on VSAM.DATA.SET.BCL, restored to whatever its original storage allocation was
VSAM.DATA.SET.P1	Path based on VSAM.DATA.SET.AIX.1, restored to whatever its original storage allocation was
VSAM.DATA.SET.P2	Path based on VSAM.DATA.SET.AIX.2, restored to whatever its original storage allocation was

Index

Using the VSE/VSAM Backup/Restore Feature (File No. S370-30) Printed in U.S.A. SC24-5216-1

Using the VSE/VSAM
Backup/Restore Feature
Order No. SC24-5216-1

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. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate. Comments may be written in your own language; English is not required.

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.)

Note: Staples can cause problems with automated mail sorting equipment.
Please use pressure sensitive or other gummed tape to seal this form.

SC24-5216-1

Reader's Comment Form

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 6R1
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®

Cut or Fold Along Line

Using the VSE/VSAM Backup/Restore Feature (File No. S370-30) Printed in U.S.A. SC24-5216-1

SC24-5216-1

Using the VSE/VSAM Backup/Restore Feature (File No. S370-30) Printed in U.S.A. SC24-5216-1