SY26-3826-3
File No. S370-30

Systems

OS/VS2 MVS Catalog Management Logic

Release 3.7

Includes Selectable Units:

Supervisor Performance #2	VS2.03.807
Data Management	VS2.03.808
Data Management Support	5752-860

IBM

Fourth Edition (January 1976)

This edition, as amended by technical newsletter SN26-0912, applies to release 3.7 of OS/VS2 and to any subsequent releases unless otherwise indicated in new editions or technical newsletters.

Significant system changes are summarized under "Summary of Amendments" following the list of figures.

Information in this publication is subject to significant change. Any such changes will be published in new editions or technical newsletters. Before using the publication, consult the latest *IBM System/370 Bibliography,* GC20-0001, and the technical newsletters that amend the bibliography, to learn which editions and technical newsletters are applicable and current.

Requests for copies of IBM publications should be made to the IBM branch office that serves you.

Forms for readers' comments are provided at the back of the publication. If the forms have been removed, comments may be addressed to IBM Corporation, Programming Publishing, P. O. Box 50020, San Jose, California 95150. All comments and suggestions become the property of IBM.

PREFACE

This book describes the internal logic of OS/VS2 catalog management and contains diagnostic information. It is directed to maintenance personnel and development programmers who require an in-depth knowledge of catalog management's design, organization, and data areas.

Organization of This Book

This book has the following major divisions:

- "Introduction," which describes catalog management and explains how catalog management interacts with the operating system.
- "Method of Operation," which describes the functions performed by catalog management.
- "Program Organization," which describes the information contained in catalog management program listings and illustrates the flow of control between modules.
- "Directory," which lists catalog management modules and the Method of Operation diagrams and Program Organization compendiums that are related to each module.
- "Data Areas," which describes control blocks used by catalog management and describes the format of catalog records.
- "Diagnostic Aids," which contains useful information for locating the cause of problems in the catalog management modules.
- "Glossary," which defines terms relevant to catalog management and lists abbreviations and acronyms used in this book and in the catalog management program listings.
- "Index," which is a subject index to the book.

Required Reading

The following books should be read and understood before using this one:

- *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide*, GC26-3838, which introduces VSAM concepts and contains definitive explanations of VSAM macro instructions.
- *OS/VS2 Access Method Services*, GC26-3841, which describes the commands used to recover catalogs and process catalog and catalog recovery area (CRA) records.

Other IBM Publications Referenced

- *OS/VS Data Management Macro Instructions*, GC26-3793
- *OS/VS Mass Storage System (MSS) Services for Space Management*, GC35-0012
- *OS/VS Mass Storage System (MSS) Services Logic*, SY35-0015
- *OS/VS Message Library: VS2 System Messages*, GC38-1002
- *OS/VS Virtual Storage Access Method (VSAM) Options for Advanced Applications*, GC26-3819
- *OS/VS2 Access Method Services Logic*, SY35-0010
- *MVS Data Areas Usage Table*, SYB8-0742
- *OS/VS2 DADSM Logic*, SY26-3828
- *OS/VS2 Data Areas*, SYB8-0606
- *OS/VS2 JCL*, GC28-0692
- *OS/VS2 Open/Close/EOV Logic*, SY26-3827
- *OS/VS2 MVS Resource Access Control Facility (RACF): General Information Manual*, GC28-0722
- *OS/VS2 System Programming Library: Service Aids*, GC28-0674
- *OS/VS2 System Programming Library: System Generation Reference*, GC26-3792
- *OS/VS2 System Programming Library: System Management Facilities*, GC28-0706
- *OS/VS2 Virtual Storage Access Method (VSAM) Logic*, SY26-3825
- *OS/VS2 System Programming Library: Data Management*, GC26-3830

Using This Book

This book is designed to be used with the catalog management program listings. The diagrams in "Method of Operation" describe the major functions performed by catalog management. These diagrams are intended to be your key to a module name (and procedure name, as appropriate) in the listing. See "Introduction" for a description of how to read these diagrams. For information on what is available in the program listings, see "Program Organization."

CONTENTS

ILLUSTRATIONS

Figures

Diagrams

SUMMARY OF AMENDMENTS

July, 1978

This technical newsletter incorporates and replaces all previous SU information in this publication.

OS/VS2 MVS Data Management Support Selectable Unit (5752-860)

A common password/bypass function is provided in SCRATCH and RENAME. HSM will use this function while copying and restoring data sets to migration storage from public storage and vice versa.

OS/VS2 MVS Data Management (VS2.03.808)

Extended CVOL Support

In OS/VS2 MVS Release 2, the VSAM catalog replaced the system master catalog and support of OS CVOLs was limited to a subset of what had been provided in OS/VS2 SVS Release 1. Extended CVOL support provides a CVOL function that is equivalent to SVS while retaining the VSAM master catalog as the only system master catalog. This support required that changes be made to Diagrams AA1, CA1, and DB1.

Improved Catalog Recovery

A brief description of the Access Method Services RESETCAT command has been added to the introduction of this publication. No technical changes were required. RESETCAT is used to reset a catalog to the level of its owned volumes without moving data, and the command is described in *OS/VS2 Access Method Services.*

OS/VS2 MVS Supervisor Performance #2 (VS2.03.807)

Resource Access Control Facility

Resource Access Control Facility (RACF) is an IBM program product that provides additional access-control measures. The Catalog Management interface to RACF has been added to this publication. Specifically:

- Change Method of Operation Diagrams DD1, DD2, DG1, EC1, EH2, and EJ3
- Change Program Organization Figures 18 and 19
- Add new Program Organization Figures 38.1 - 38.9 to describe Catalog Management RACF functions
- Add a security flags field to the Data and Index, Cluster, Alternate Index, and Path Catalog Records
- Add new fields and bit settings in the ACB, the CAXWA, the CTGFV, and the CTGPL

Catalog RPL Limit

Catalog RPL Limit is an improved method of handling catalog RPLs. An incoming request unable to obtain an RPL now will issue the WAIT macro for a free RPL. When an in-process request frees an RPL, it will issue a cross-memory POST macro on the waiting request's ECB. Changes to support this function include:

- New fields and bit settings in the CAXWA and the CCA
- IGGPACI, IGGPCDSO, IGGPPCCP, and IGGPSC2 have been moved from IGG0CLA3 to IGG0CLAH
- IGGPCSCP, IGGPCVX, IGGPDOG, and IGGPDPCC are now designated external procedures in IGG0CLA3

Support for ASM Rewrite

The support for ASM rewrite allows the defining and preformatting of Swap spaces and treats SYS1.STGINDEX as a system data set. The changes are:

- A new bit setting in the ATTR2 field of the Data and Index Catalog Record describes an internal system data set
- A new bit setting in the CATTR field of the Cluster Catalog Record indicates a cluster that describes a SWAPSPACE

Release 3.7

Recoverable Catalog Support

Catalogs can be defined with an optional recovery attribute that allows data sets to be recovered or restored. Recovery is based on information that is recorded in the catalog and also in a catalog recovery area on the volumes owned by the catalog.

Recoverable catalog support changes the format of the catalog record, and adds a new Method of Operation diagram, a new catalog record, and new fields to various catalog control blocks.

Catalog Unload/Reload

Catalog Unload/Reload, a part of the Access Method Services REPRO command, is a high-performance option used to back-up VSAM catalogs that do not have the recoverable attribute.

Enhanced VSAM

Extensive changes have been made throughout this publication to describe Catalog Management's support of Enhanced VSAM's functions and data structures, including alternate indexes, spanned records, relative record data sets, and reusable data sets.

Most Method of Operation diagrams have been changed; new diagrams describe the catalog processing required when (1) an alternate index or path is defined or deleted, (2) the VSAM object to be processed is reusable, and (3) a SHOWCAT macro instruction is issued.

All Program Organization figures have been changed. Flowcharts describing catalog and catalog recovery area I/O functions have been given figure numbers to make then easier to identify and locate.

The Data Areas section includes four new catalog record types: alternate index, path, upgrade, and catalog recovery area records. Changes have been made to most control blocks to accomodate the new functions and structures.

Release 3

- The "Method of Operation" section contains a description of CONVERTV processing for converting data stored on a direct-access storage volume to a mass storage volume, or *vice versa,* with the IBM 3850 Mass Storage System.
- The "Method of Operation" section contains a description of ALTER REMOVEVOLUMES processing for taking away VSAM ownership of a volume without gaining access to the catalog that owns the volume.
- Other sections contain miscellaneous information: program organization description for CONVERTV, directory entries, data areas, messages, macros, error codes.

Release 2

- *OS/VS2 (VSAM) Catalog Management* replaces *OS/VS Catalog Management* as the system catalog management component. The "Introduction" describes the interactions between OS/VS catalog management and various system components.
- The "Introduction" and "Method of Operation" sections include descriptions of enhancements to VSAM catalog management—aliases, generic names, generation data groups, and copying a catalog.

INTRODUCTION

OS/VS2 catalog management uses VSAM to read and write catalog records on a direct-access storage device. Virtual Storage Access Method (VSAM) is an access method for use with OS/VS2. VSAM is used with direct-access storage to provide fast storage and retrieval of data.

Catalog records are built and stored in the VSAM record format. VSAM's record format is different from that of other access methods. All VSAM records are stored in *control intervals*. A control interval is a continuous segment of auxiliary storage. The records are ordered according to values in a key field (key-sequenced), when they were stored (entry-sequenced), or where they were stored (relative record). With key-sequenced data sets, the user can gain access to a record by specifying either its key or its relative byte address (RBA). With entry-sequenced data sets, the user can gain access to a record only by specifying its RBA. With relative record data sets, the user can gain access to a record by specifying its relative record number. For additional information on VSAM records and how they are stored, see "Data Areas" in *OS/VS2 VSAM Logic*.

OS/VS2 catalog management resides in the pageable link pack area. Figure 1 illustrates VSAM's relationship to OS/VS, the processing program, and the data stored on a direct-access storage device and in mass storage.

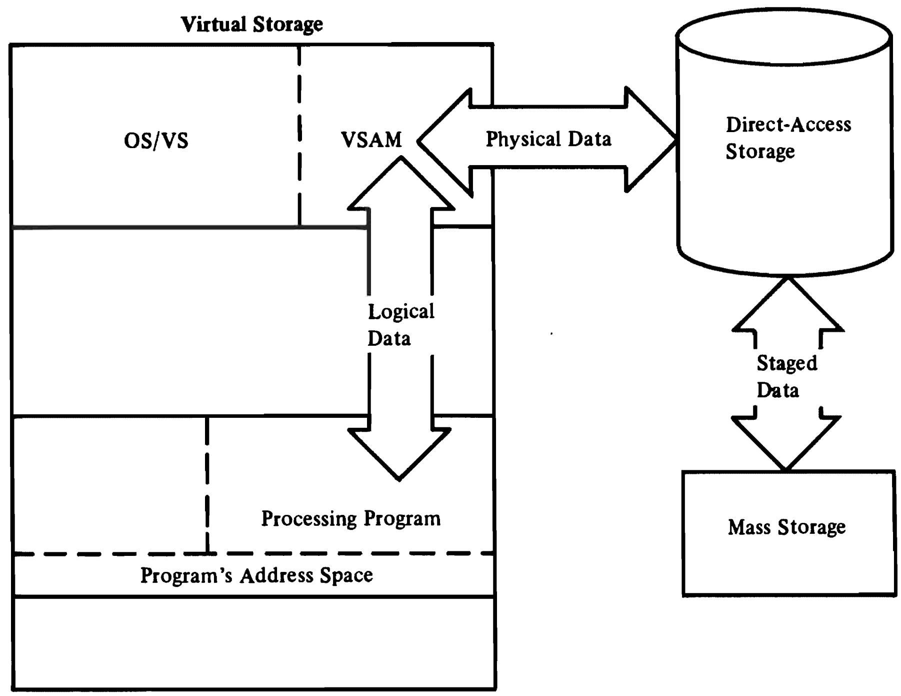


Figure 1. Relationship of VSAM, OS/VS, User's Processing Program, and Staged Data

VSAM is controlled by user macro instructions. For information on VSAM user macro instructions, see *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide* and *OS/VS Virtual Storage Access Method (VSAM) Options for Advanced Applications*.

VSAM communicates with other parts of the operating system through the SVC processor and through OS/VS control blocks used by VSAM. In addition to the OS/VS control blocks used by VSAM, VSAM builds and uses the access method control block (ACB). The ACB describes a VSAM data set in much the same way that a DCB describes a nonVSAM data set. See *OS/VS2 VSAM Logic* for a more detailed description of the VSAM access method.

In addition to processing records and data sets, VSAM opens and closes data sets and does most of its own space management. That is, VSAM makes only minor use of OS/VS Open and Close and relies on OS/VS DADSM (direct-access device space management) for only part of its space management. To do much of this work, VSAM uses the OS/VS2 master catalog and user catalogs. OS/VS2 catalogs contain a description of VSAM space—where available space is and how space is used—and the location of data sets.

Using Catalog Management

OS/VS2 catalog management is used to locate and update information in an OS/VS2 catalog. When an OS/VS component requests information from an OS/VS2 catalog, the component builds a CTGPL (catalog parameter list) to describe the request and CTGFLs (catalog field parameter lists) to describe specific fields of information needed to satisfy the request. The component then issues CATLG, a macro instruction that results in SVC 26, with register 1 pointing to the CTGPL. Only OS/VS components and Access Method Services issue CATLG. User programs have no direct interface with catalog management.

Existing user programs issue CAMLST to build an OS/VS catalog processing parameter list, then issue SVC 26 to call OS/VS catalog management. The catalog controller interprets the CAMLST and builds a CTGPL and CTGFLs (see Diagram CA1, Catalog Controller Processing, for details.) The user's program can also build a CTGPL and CTGFLs, then issue SVC 26, to process catalog record information.

JOBCAT and STEPCAT DD Statements

The master catalog is assumed to contain the definition of the data set described in a DD statement if no user catalog is indicated or if the definition is not found in the user catalog(s) that are indicated. A user catalog can be specified for all steps of a job (with a JOBCAT DD statement) or for a particular job step (with a STEPCAT DD statement).

The order in which OS/VS2 catalog management searches catalogs for a data set is:

1. Any user catalog(s) specified for the current job step.
2. Any user catalog(s) specified for the current job, only if no user catalogs are specified for the job step.
3. A dynamically opened catalog, identified by
   - catalog DSNAME, or
   - first qualifier of a data set's qualified DSNAME
4. The master catalog.

If a job step contains no STEPCAT DD statements, but the job contains a JOBCAT DD statement, the OS/VS Scheduler designates the JOBCAT catalog(s) as STEPCAT catalog(s) for the job step.

OS/VS Scheduler

The OS/VS Scheduler analyzes the user's job control language statements. When a JOBCAT or STEPCAT DD statement is found, the user has specified a catalog that contains information about a data set his job intends to access. The OS/VS Scheduler opens each catalog specified by the user before allowing the user's program to begin processing. When the user's program completes processing, the OS/VS Scheduler closes each catalog opened for the user.

OS/VS VSAM Open/Close/End-of-Volume

VSAM Open: When a user's program issues the OPEN macro instruction to open a VSAM data set (each VSAM data set is cataloged in an OS/VS2 catalog), VSAM Open processing calls OS/VS2 catalog management to obtain information about the data set. If the data set is a key-sequenced VSAM data set, for example, the information returned by catalog management includes: location of each extent of the data set and its index, attributes of the data set, data set statistics (the contents of the AMDSB control block when the data set was last closed), key range information, and password information. If the data set is a nonVSAM data set, OS/VS2 catalog management returns the volume serial number and device type of the volume containing the data set.

VSAM Close: When a user's program issues the CLOSE macro instruction to close a VSAM data set (cataloged in an OS/VS2 catalog), VSAM Close processing calls OS/VS2 catalog management to update the data set's statistics in the catlaog.

VSAM End of Volume: VSAM End of Volume processing is called by VSAM Record Management when either of the following occurs:

- A write operation failed because there is no more space allocated to the VSAM data set.
- A read operation failed because the desired record is on an unmounted volume.

VSAM End of Volume processing calls catalog management to obtain additional space or to obtain the volume's serial number and characteristics. When additional space is obtained for the data set, catalog management updates records that describe the data set and that describe the volume containing the space.

OS/VS Access Method Services

Access Method Services is a service program that allows VSAM users to catalog and uncatalog VSAM data sets. Access Method Services commands that result in a direct call to OS/VS2 catalog management are:

DEFINE	Catalog an object in an OS/VS2 catalog
ALTER	Modify information in a catalog record
DELETE	Uncatalog an object from an OS/VS2 catalog
LISTCAT	Print catalog information
CONVERTV	Convert a volume to or from mass storage

DEFINE is used to build a catalog record (or group of catalog records) that describes the object—a VSAM cluster, a nonVSAM data set, an alternate DSNAME, a generation data group, a VSAM data space, an OS/VS2 catalog—being created or cataloged.

ALTER is used to modify information within certain types of catalog records.

DELETE is used to uncatalog an object that is described by a catalog record. If the object has space allocated to it, the space is returned to the OS/VS system.

LISTCAT is used to obtain a printout of information contained in caller-specified types of catalog records.

CONVERTV is used to convert a direct-access storage volume to a mass storage volume, or *vice versa*. That is, data that is stored on one type of volume is stored on the other type, and catalog volume information is updated to record the change. For additional information on this command, see *OS/VS2 Access Method Services* and *OS/VS Mass Storage System (MSS) Services for Space Management.*

OS Catalog Management

Certain commands issued for OS catalog management are converted into OS/VS2 catalog management requests and processed against OS/VS2 catalog records.

Creating the OS/VS2 Master Catalog

The OS/VS2 master catalog is created during the system generation process (SYSGEN). The master catalog is identical to a user's catalog, except that it is identified by a record in the SYS1.NUCLEUS data set. See Diagram BA1, Opening the Master Catalog, for details about the relationship of the master catalog to the OS/VS2 system.

When SYSGEN occurs, the OS/VS2 master catalog is created. The SYSGEN process builds three series of job steps that build the master catalog:

1. The Access Method Services command DEFINE CATALOG is issued to build an OS/VS2 catalog. See Diagram EE, DEFINE CATALOG: Create an OS/VS2 Catalog, for details on the catalog creation process.

   The SYSGEN starter system includes a master catalog, called the functional system master catalog. The master catalog for the system being generated is called the target system master catalog, and is pointed to by a User Catalog catalog record in the functional system master catalog.

2. All system data sets are defined in the catalog.

   A. The Access Method Services command DEFINE NONVSAM is issued to define each nonVSAM data set. See Diagram EF1, DEFINE NONVSAM: Define a NonVSAM Data Set In an OS/VS2 Catalog, for details on the define nonVSAM process. The system data sets defined in the catalog are:

SYS1.BRODCAST	SYS1.LOGREC	SYS1.SAMPLIB
SYS1.CMDLIB	SYS1.LPALIB	SYS1.SVCLIB
SYS1.DCMLIB	SYS1.MACLIB	SYS1.SYSJOBQE
SYS1.DSSVM	SYS1.MANX	SYS1.SYSVLOGX
SYS1.DUMP	SYS1.MANY	SYS1.SYSVLOGY
SYS1.HELP	SYS1.NUCLEUS	SYS1.TELCMLIB
SYS1.IMAGELIB	SYS1.PARMLIB	SYS1.UADS
SYS1.LINKLIB	SYS1.PROCLIB	SYS1.VVIC

   B. The Access Method Services command DEFINE CLUSTER or DEFINE PAGESPACE is issued to define each VSAM data set or paging space. See Diagram ED, DEFINE CLUSTER: Create a Cluster, for details on the Define Cluster process. The VSAM system data sets defined in the catalog are:

      SYS1.STGINDX
      Paging spaces

   C. If the user provided DSNAMEs of nonVSAM and VSAM data sets to be included in the OS/VS2 master catalog, the DEFINE NONVSAM and DEFINE CLUSTER commands are issued to define each user-specified data set.

3. The SYS1.NUCLEUS data set (in the nucleus being generated) is updated with a record, labelled "SYSCATLG", that describes the OS/VS2 master catalog's location—its volume serial number. The catalog volume's VTOC contains a DSCB identified by the name "Z9999994.VSAMDSPC. Taaaaaaa.Tbbbbbbb", where "a" and "b" are digits of the timestamp. The DSCB describes the space containing the first extent of the catalog's low-key range. This part of the catalog contains the catalog's self-describing records in the first fourteen control intervals. See "Catalog Records That Describe the Catalog" for a description of the self-describing catalog records. See "Catalog" for a description of the OS/VS2 catalog. Both of the above references are in the "Introduction," and immediately follow.

For further information on system generation and the OS/VS2 master catalog creation process, see *OS/VS2 System Programming Library: System Generation Reference.*

Catalog

OS/VS2 catalogs—the master catalog and any user catalogs—are built and processed by catalog management modules. Catalog management modules, via the catalog, enable a user to locate a data set, volume, index, or cluster by specifying a data set's dsname or volume serial number. In addition, OS/VS2 catalogs provide VSAM with the information required to allocate space for data sets, verify authorization to gain access to data sets, compile usage statistics on data sets, and relate RBAs to physical locations within data sets. The catalog indicates, therefore, much more than the simple location of data sets. The catalog maintains the relationship between a key-sequenced data set and its index, describes the location of VSAM data spaces and the data sets that reside in them, and describes the space that is available for new data sets.

The catalog is conceptually a key-sequenced VSAM data set divided into two key ranges called the low-key range and the high-key range. VSAM data set processing options, such as index record replication and sequence set with data, are utilized in both parts of the catalog. The catalog record size is variable; the catalog control interval size is 512 bytes. Figure 2 shows an OS/VS2 catalog. The figure shows:

- The low-key range of the catalog, shown on the left, contains records that describe generation data groups, alias names, and objects—data sets, indexes, alternate indexes, paths, upgrade sets, volumes, and clusters.
- The high-key range of the catalog, shown on the right, contains the true name (a data-set name, cluster name, or volume serial number) of an object specified by the user.
- The index, shown in the middle, points to both the low- and high-key ranges of the catalog.

With the exception of catalog records that are built when the catalog is created and describe the catalog itself, catalog records are built whenever objects are cataloged. The order that records are in depends upon which portion of the catalog the records belong to. The catalog records that reside in the low-key range of the catalog are ordered according to control interval number. As objects are cataloged, available control intervals are used. The catalog records that reside in the high-key range of the catalog are ordered according to their true name (cluster dsname or volume serial number).

Catalog management relies on VSAM record management for all record retrieval and storage. When a user specifies a data set name, for example, record management uses the index to retrieve a catalog record that contains the data-set name (in the high-key range of the catalog); that record, in turn, contains the control-interval number of the catalog record that describes the data set. Catalog management converts the control-interval number to an RBA in the low-key range of the catalog.

High-Key Range of the Catalog

The high-key range of the catalog contains 47-byte True Name records in 512-byte control intervals. The True Name records associate user-specified names or volume serial numbers with the control-interval number of the catalog record that describes the specified object.

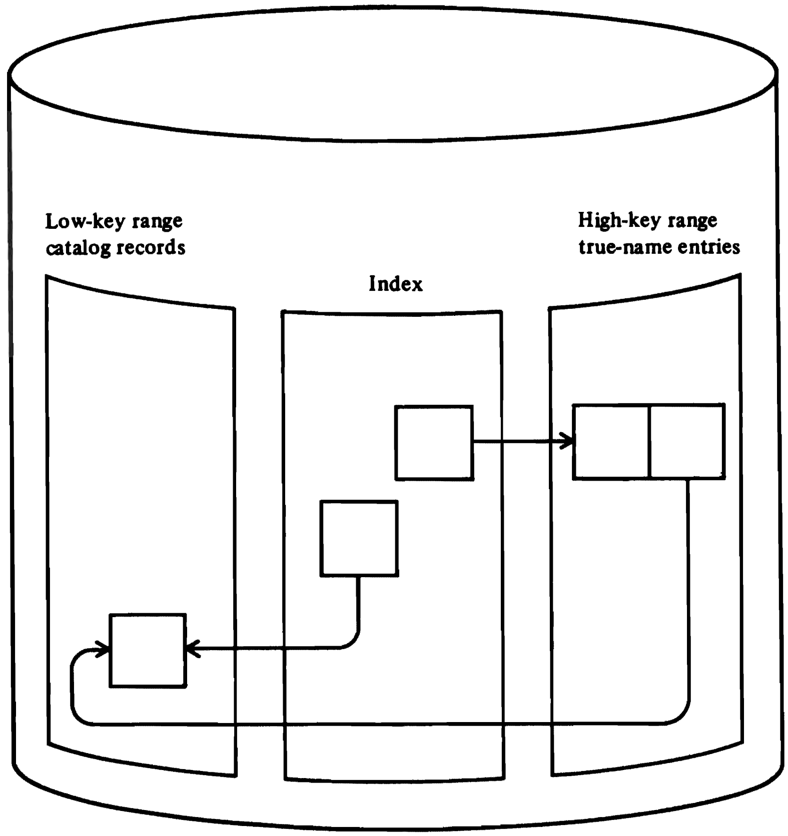


Figure 2. Parts of an OS/VS2 Catalog

Low-Key Range of the Catalog

Records in the low-key range are 505 bytes long. Each record resides in its own control interval. Each record also contains its record type. The low-key range of the catalog is made up of the following types of records:

- Alias record, which relates an alternate DSNAME (alias) of a nonVSAM data set or user catalog to the data set's or user catalog's catalog record. This record is record type "X."
- Alternate index record, which relates the alternate index to its associated base cluster and also to any paths over it. This record is record type "G."
- Cluster record, which describes a VSAM data-set cluster or pagespace (conceptually an entry-sequenced VSAM data set). This record contains the control-interval number of a Data record and, if the VSAM data set is a key-sequenced data set, the control-interval number of an Index record. There is one Cluster record for each VSAM cluster cataloged. This record is record type "C."
- Control record, or CCR, which describes the free control intervals in the low-key range of the catalog. The Control record is the fourth record in the catalog. This record is record type "L."
- Data and Index records, which describe data sets and indexes. A Data record can also describe a pagespace (conceptually an entry-sequenced VSAM data set). There is one Data or Index record for each data set or index cataloged. These records are record types "D" and "I."

- Extension record, which contains overflow information from another catalog record. There are as many Extension records as are required to contain overflow information. This record is record type "W" when it is an extension of a Volume record; it is record type "E" when it is an extension of any other catalog record.
- Free record, which marks the control interval in which it resides as available for use as another kind of catalog record. There is one Free record for each previously assigned control interval that is available for use. This record is record type "F."
- Generation data group base record, which contains the control interval number and generation level (absolute name) of each generation data set in the group. This record is record type "B."
- NonVSAM record, which describes a nonVSAM data set. There is one NonVSAM record for each nonVSAM data set cataloged. This record is record type "A."
- Path record, which relates a base cluster and possibly an alternate index. This record is record type "R."
- Upgrade set record, which relates the data components and index components of the alternate indexes that comprise the upgrade set. This record is record type "Y."
- User-Catalog record, which describes a VSAM user catalog. There is one User-Catalog record for each user catalog connected to this master catalog. This record is record type "U."
- Volume record, which describes each VSAM data space on a volume, the data sets that reside in the data space, and the space available within the data space. There is one Volume record for each volume controlled by this catalog. This record is record type "V."

The Cluster, Data, Index, Alternate Index, Path, Upgrade Set, NonVSAM, Extension, Alias, GDG Base, and User-Catalog records have a common general format. See the "Data Areas" section for a figure that shows the general format of these records.

Catalog Records That Describe the Catalog

Catalog records that describe the catalog as a data set are in fixed positions at the beginning of the catalog. The following table shows the control-interval numbers of records that describe the catalog, the kind of catalog record each is, and the contents of each.

Control Interval Number	Record Type	Contents
0	Data Set	Description of the data portion of the catalog (low-key range and high-key range).
1	Index	Description of the index portion of the catalog.
2	Cluster	Description of the catalog as a key-sequenced VSAM cluster. This catalog record contains the catalog's password information set of fields.
3	Control	Catalog control record (CCR), which describes the catalog's free control intervals within the low-key range.
4	Extension	Extension of the index catalog record (control interval number 1). This Extension record contains a description of the high-level index extents of the catalog.
5	Extension	Extension of the data set catalog record (control interval number 0). This Extension record contains a description of the low-key range data extents of the catalog.
6	Extension	Extension of the index catalog record (control interval number 1). This Extension record contains a description of the index sequence-set extents for the low-key range of the catalog.
7	Extension	Extension of the data set catalog record (control interval number 0). This Extension record contains a description of the extents of the True Name records in the high-key range of the catalog.
8	Extension	Extension of the catalog's index catalog record (control interval number 1). This Extension record contains a description of the index sequence-set extents for the high-key range of the catalog.
9	Volume	Description of the track allocation and VSAM data spaces on the volume containing the catalog.
10-13	Volume Extension	As many volume extension records as are necessary to describe the total space on the volume. These extension records contain the volume record's space map set of fields.

When the catalog is built, there are two True Name records. One contains the catalog's volume serial number and points to control interval number 9. The other contains the catalog's dsname and points to control interval number 2.

Alternate Names (Aliases)

An *alias* is an alternate DSNAME for a nonVSAM data set's or user catalog's DSNAME. Any of the 44 characters of the data set or catalog name can be different in the alias. The user creates an alias by using the Access Method Services DEFINE command to build an alias-name catalog record that contains the alias and points to the data set's (or catalog's) catalog record. No passwords are required to modify or delete alias catalog records. The alias user-catalog name capability allows many users to refer to the same catalog—each user can specify a different alias user-catalog name.

Generation Data Groups

A *generation data group* (GDG) is a collection of nonVSAM data sets that have a common external name, and that are related by the time sequence in which they are cataloged (that is, their generation). For example, a data set named LAB.PAYROLL(0) refers to the most recent data set of the group. LAB.PAYROLL(−1) refers to the second most recent, etc. The number in parentheses is called the *relative number* of a specific generation.

When a user locates a generation data set by specifying a relative DSNAME (dsname(n)), catalog management replaces the relative number (n) with an *absolute name*. The absolute name is the simple name of the data set. The absolute name has the general form G*nnnn*V*mm*, where *nnnn* is the generation number and *mm* is the version number. If the relative number is zero, the last generation data set cataloged is retrieved. If the relative number is minus one, the next to last generation data set cataloged is retrieved. If the relative number is a positive number, catalog management will compute the absolute name by adding the user-specified relative number to the absolute name of the last (highest) generation cataloged. The user can also specify the absolute name to be used as a reference instead of the last (highest) generation cataloged.

The user creates a generation data group by using the Access Method Services DEFINE command to build a generation data group base catalog record. There is one GDG base catalog record in the catalog for each GDG. Each GDG base catalog record contains a list of absolute names in sequential order for those generation data sets which are members of this group. Since the OS/VS2 catalog does not contain an OS catalog index structure to support GDG processing, the user must issue the Access Method Services DEFINE command to create each GDG base record.

Password Protection

VSAM passwords control the right to use VSAM data sets. VSAM passwords restrict unauthorized usage of OS/VS2 catalogs, clusters, data sets, and indexes. If an OS/VS2 catalog or cluster is protected by passwords, the passwords are contained in a password set of fields in the cluster, data set, alternate index, path, or index catalog record.

Passwords are initially defined when the user issues the Access Method Services DEFINE command to create a catalog or VSAM cluster. The user can issue the Access Method Services ALTER command to change a catalog's or cluster's passwords when the user supplies the catalog's or cluster's master password.

When an unauthorized user attempts to access a password-protected catalog or cluster and is unable to supply the correct password, catalog management sets an error return code in the user's ACB and writes an SMF (System Management Facilities) security violation record.

A user can specify additional security verification by defining a USVR (User Security Verification Routine) when the catalog or cluster is created. If a USVR is specified for a catalog or cluster, the USVR is processed when the catalog management security verification processing completes sucessfully.

A *privileged* user of catalog management bypasses security verification. Privileged users include the OS/VS Scheduler and OS/VS Open/Close/EOV processing. OS/VS Checkpoint/Restart is not privileged and does not bypass security verification.

Levels of Password Protection

The password set of fields contains four levels of password protection: master, control interval, update, and read-only. If any level of password is null (the password doesn't exist), operations requiring that level of password protection can be performed without requiring the user to supply a password. When the user supplies a password for one level, any operation requiring that level *and* lower levels of password protection can be performed.

Master Password

The master password is the highest in the password hierarchy. The catalog's master password is required to modify or delete any password-protected catalog record, to retrieve any password in the catalog, or to open the catalog as a data set. Since all passwords reside in OS/VS2 catalogs, a *password-protected* catalog record is a VSAM cluster, data set, index, alternate index, or path catalog record that contains a password set of fields. If the user supplies a cluster's master password, his program can modify the data set and index catalog records and retrieve their passwords.

Control Interval Password

The control interval password is the second highest level in the password hierarchy. The control interval or master password of a password-protected VSAM cluster, data set, index, alternate index, or path is required when the user opens the object for control interval processing.

Update Password

The update password is the third password in the password hierarchy. The catalog's update or higher level password is required when the user issues an Access Method Services DEFINE command to create a VSAM object and/or add a catalog record to the catalog. Even though the new catalog record doesn't contain a password set of fields, the user must still supply the catalog's update or higher password. The catalog's update or higher level password is also required to delete a volume catalog record. The cluster's update or higher level password is required to open a password-protected VSAM cluster for update processing.

Read-Only password

The read-only password is the lowest level in the password hierarchy. The catalog's read-only or higher level password is required when the user issues an Access Method Services LISTCAT command to retrieve catalog record information, except passwords. The catalog's master password is required to retrieve password information. The cluster's read-only or higher level password is required when a user opens the cluster for read-only processing. When the user supplies an OS/VS2 catalog record's read-only password, his program is allowed to retrieve information (except passwords) from the record. The record's master password is required to retrieve password information.

Restrictions

Only cluster, data set, index, alternate index, and path catalog records can contain a password set of fields. The catalog's password set of fields is in control interval number 2—one of the catalog's self-describing records. The other types of catalog records—nonVSAM, volume, user catalog, upgrade, GDG base, and alias name—do not contain a password set of fields.

No password is required to:

- Alter or delete a catalog record that doesn't contain a password set of fields, except that the catalog's update password is required to delete or alter nonVSAM catalog records.
- Open a cluster described by a catalog record that doesn't contain a password set of fields.
- Add, alter, or delete catalog records in a catalog that is not protected by passwords.

When a user issues LOC, an OS Catalog Management command, against a nonVSAM catalog record in an OS/VS2 catalog, no password is required. If the OS/VS2 catalog is protected by an update level password, the update level password or higher level is required when the user issues the CAT, UNCAT, or RECAT command.

When IEHDASDR, an OS Utility program, attempts to access records in a VSAM data space, the user must supply the read-only password for each read-only password-protected data set in the data space.

RACF Protection

Resource Access Control Facility (RACF) provides software access control measures that you can use in addition to your present data security measures (such as TSO passwords or password-protected data sets).

When RACF protection and password protection are both applied to a data set, password protection is bypassed, and use is authorized solely through the RACF checking system.

RACF is described in *OS/VS2 MVS Resource Access Control Facility (RACF) General Information Manual.*

Copying the Catalog

A user can copy a catalog from one direct-access device to another. The original catalog is called the "source" catalog. The new catalog is called the target catalog. The target catalog contains volume catalog records that describe the target catalog volume's direct-access device space. When the source catalog is copied into the target catalog, the target catalog contains volume catalog records that describe the source catalog volume's direct-access device space.

OS/VS2 Access Method Services describes the REPRO command, which is used to copy the catalog. However, the catalog copying procedure is more involved than issuing the REPRO command:

1. The Access Method Services DEFINE CATALOG command is used to create a target catalog. The target catalog is created on a direct-access volume that contains no VSAM data spaces.

2. The Access Method Services REPRO command is issued to move each catalog record from the source catalog into the target catalog.

   Source catalog records are moved into the target catalog starting at the target catalog's 14th control interval. Each source catalog record is scanned to find all referenced control interval numbers. Each control interval number in the source catalog record is updated (control interval number + 13) to reflect the referred-to record's new location in the target catalog.

   IDACAT21 is called by the Access Method Services REPRO processing module to update the source catalog record's control interval number references and to move the updated record into the target catalog.

3. The Access Method Services EXPORT DISCONNECT command is issued to remove the source catalog's User Catalog catalog record from the OS/VS2 master catalog. The source catalog can no longer be opened as a user catalog.

4. The Access Method Services DELETE CLUSTER command is issued to delete the source catalog's self-describing records in the target catalog. The source catalog is described in the target catalog as a key-sequenced VSAM data set (control intervals 14 through 22 in the target catalog). Catalog management also modifies the source catalog volume's VTOC so that the format-1 DSCB that pointed to the source catalog's VSAM data space indicates that the data space doesn't contain a catalog.

   The source catalog is effectively removed from the OS/VS2 system. Its space is marked "unallocated" in the target catalog and can be used to contain other data.

Recoverable Catalog Support

Catalogs can be defined with an optional recovery attribute that allows data sets to be recovered or restored. Recovery is based on information that is recorded in the catalog and also in a catalog recovery area on the volumes owned by the catalog. The recovery area is established when the recoverable catalog acquires ownership of the volume. Thus, all volumes owned by a recoverable catalog contain catalog recovery area space.

Whenever records in a recoverable catalog are defined, deleted, or modified, the corresponding information in the catalog recovery area is updated to reflect the change. Although no specific commands are required to maintain the recovery area, certain volumes must be mounted during defines, alters, deletes, and any catalog entry modifications resulting from open, close, or end of volume activity. The volumes are:

ALTER The prime catalog recovery volume for the objects being altered.

DELETE All volumes that are referenced by the entry being deleted and the prime CRA volume.

DEFINE All volumes that are referenced in the DEFINE command. Also, the first volume of the base cluster must be mounted when alternate indexes and paths are being defined.

Once a recoverable catalog is defined, it cannot be made nonrecoverable. Also, a recoverable catalog cannot be copied. A nonrecoverable catalog can be converted and made recoverable through Access Method Services commands. This conversion is not necessary unless there is a requirement for the recovery capability.

The Access Method Services commands used to achieve catalog recovery are described in *OS/VS2 Access Method Services*. That book also contains specific instructions on how to make an existing master catalog recoverable. User catalogs can be converted by using the EXPORT, DELETE, DEFINE, and IMPORT commands.

This publication contains additional information about catalog recovery. In the "Method of Operation" chapter, Diagram EE3 describes the processing required to define a catalog recovery area. In the "Program Organization" chapter, detailed drawings in the "Catalog Management I/O Functions" section illustrate CRA I/O operations. The "Data Areas" chapter contains the description and the format of all the recovery area records.

Resetting a Catalog

The Access Method Services RESETCAT command is used to reset a catalog to the level of its owned volumes without moving data. It merges the catalog records from specified catalog recovery areas with the records in the catalog. If the equivalent record exists in the catalog, it will be replaced in the catalog. If the equivalent record does not exist in the catalog, it will be inserted into it. If a record exists in the catalog which designates one of the specified CRA volumes as its CRA and no equivalent record exists in the CRA, that record will be deleted.

RESETCAT is described in *OS/VS2 Access Method Services.*

Catalog Unload/Reload

Catalog unload/reload is a high-performance option used to backup and recover VSAM catalogs that do not have the recoverable attribute. (This function is not used with recoverable catalogs because CRA records are not modified during the reload process, and there are no assurances that CRA records and catalog records would be synchronized.) This option is part of the Access Method Services REPRO command.

Catalog backup (unload) is performed by copying a VSAM catalog to a sequential access method (SAM) data set or a VSAM key-sequenced or entry-sequenced data set. This unloaded form of the catalog is available for subsequent reloading, but it cannot be used as a catalog while it is in the unloaded form. Unloading a catalog creates a data set that contains all of the catalog records as they existed at the time of the backup operation.

Catalog recovery (reload) is performed by copying an unloaded catalog into an existing VSAM catalog. The existing catalog can be a newly defined catalog or it can be an earlier or later version of the unloaded catalog. A new catalog (same name, volume serial number, and device type) would be defined whenever the old catalog could not be opened for processing. An earlier or later version of the unloaded catalog might be obtained via a restore of the catalog volume. In both cases the reload operation creates a catalog equivalent to the catalog that was unloaded.

The reload operation replaces any existing entries in the target catalog with entries from the backup copy. Entries in the backup catalog that are not on the target catalog are added. Entries that exist only in the target catalog are deleted.

Password protection of VSAM catalogs is optional. When a catalog is password-protected, the unload/reload option requires that the master password be provided for the catalog. The password can be supplied in the REPRO command or through the system console or TSO terminal.

The frequency of catalog backup operations will determine the degree of success in any recovery operation. Frequent backup operations can be used to ensure that the latest backup copy closely matches the active catalog and reflects the data and its allocation on the volumes owned by the catalog. A wide difference between the backup catalog and the active catalog will increase the amount of manual intervention required to regain access to all of the data.

For complete information about catalog unload/reload, see *OS/VS2 Access Method Services*.

Error Recovery

Catalog management branches to the ESTAE service routine to establish an ESTAE environment. When an error condition occurs, the catalog management recovery routine (IGG0CLA9) gets control from the System Recovery Termination Manager.

The catalog management recovery routines receive control from System Recovery Termination management when the following types of errors occur: program checks, restart key interrupts, protection check of a user-supplied address, machine checks, and PSW restart. The recovery routines release all virtual storage that was obtained for the current request. OS/VS2 catalog management (IGG0CLA1) branches to the ESTAE routine. The ESTAE routine returns to the Catalog Controller, which issues SETRP to record the conditions for error analysis.

METHOD OF OPERATION

A method of operation diagram describes one of the OS/VS2 catalog management functions by listing the process steps required to complete the function, and by showing the data required for each process step and the data produced by each process step.

Reading Method of Operation Diagrams

Method of operation diagrams are functional descriptions of catalog management. The diagram and descriptive notes, keyed to the diagram, are on facing pages.

The diagrams contain three blocks of information: input, processing, and output. The left-hand side of the diagram shows the data that serves as input to the processing steps in the center of the diagram, and the right-hand side shows the data that is output from the processing steps. Input is anything a program function refers to or gets. Processing is the steps required to fulfill the function represented by the diagram. Output is any change effected by a function; for example, register contents, or control blocks created or modified. The processing steps are numbered; the numbers correspond to notes on the facing page. The notes include cross-references to the listings. Figure 3 shows a method of operation figure.

The processing portion of the diagram shows the processing steps required to fulfill the function described by the diagram. Note that the function described by one diagram might be performed by one or more catalog management

Diagram AA1. Catalog Management Overview

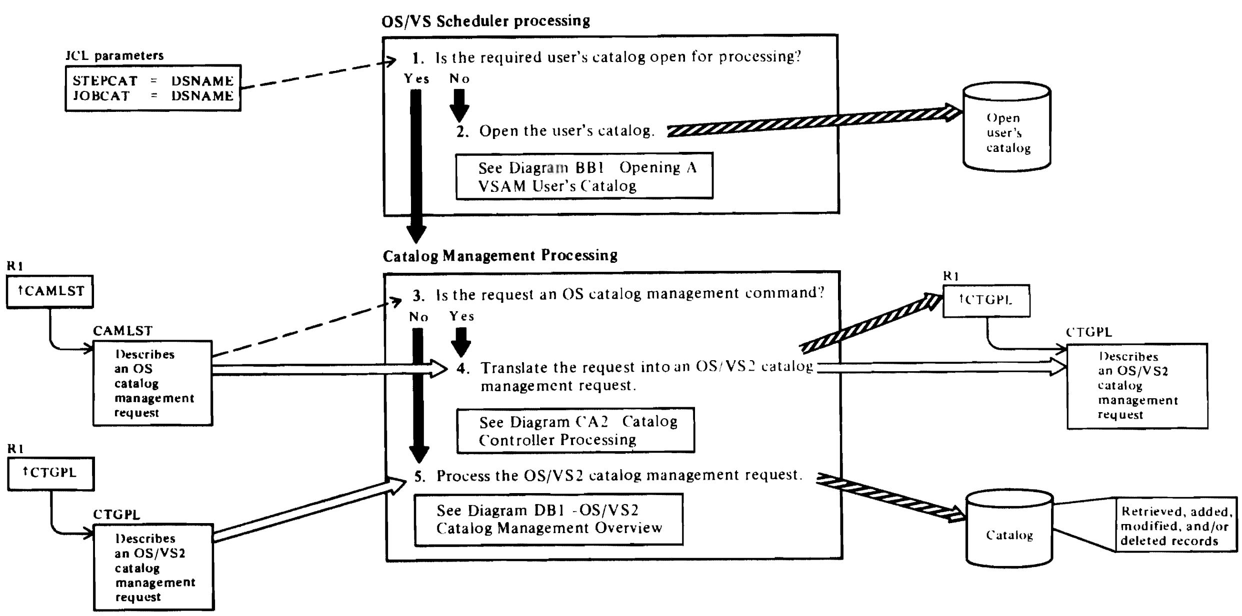


Figure 3. Method of Operation Diagram

modules; that is, the diagrams describe functions, not physical parts of the program.

The output created by each processing step is shown in the diagram.

Reading the method of operation diagrams requires that you understand the symbols they use. Figure 4 shows the symbols and describes their meaning.

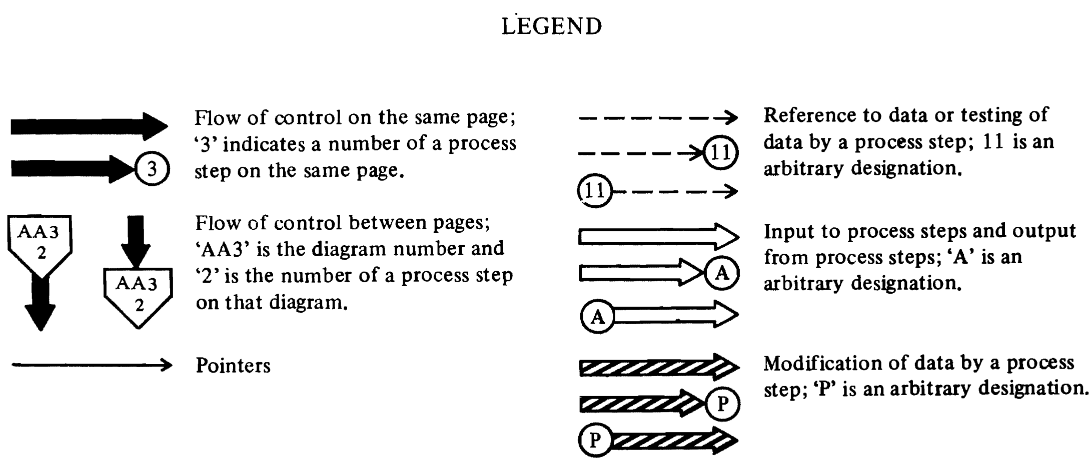


Figure 4. Graphic Symbols Used in Method of Operations Diagrams

Figure 5 shows part of the notes to Figure 3.

Notes for Diagram AA1

Catalog management is called when information in one of the catalogs available to the caller is needed to open, extend (obtain more space for), or close a data set. Catalog management is also called when information must be added to, deleted from, or modified within one of the catalogs.

1 OS/VS Scheduler

2 IEFAB4F5

When the user's JCL DD JOBCAT or STEPCAT statements specify a user's (private) catalog, the OS/VS2 Scheduler calls IEFAB4F5 to open the user's catalog. See "Open Catalog Control Blocks," a figure in the "Data Areas" section that shows how the TCB, JSCB, and PCCBs are related to identify each open catalog available to the user's job.

3 IGC0002F

If the user has issued an OS catalog management request and has specified a CVOL volume serial number in the parameter list, then IGC0002F exits to the CVOL Processor, IGG0CLCA, to process the request in the specified CVOL (see *OS/VS2 CVOL Processor Logic*).

4 IGC0002F

When the user issues an OS catalog management command without a CVOL volser, the command is translated to a OS/VS2 catalog management request and is processed by OS/VS2 catalog management routines. When the processing completes, the return code (and other information for the user's program, as applicable) is translated to an OS catalog management return code.

Note: OS catalog management requests usually apply to NonVSAM, Alias, and GDG Base catalog records.

5 IGG0CLA1

When the user is an Access Method Services routine, a part of OS/VS2 catalog management, called "catalog management services", processes the request. Catalog management services routines are described in Method of Operations Diagrams EA1 through EN1

When the user is a VSAM Open, Close, or End of Volume routine, OS/VS2 catalog management routines process the request. These routines are used to retrieve, add to, or update the information in catalog records. Catalog management services routines call the catalog management routines to manipulate catalog record information. Method of Operations Diagrams DA1 through DM3 describe the OS/VS2 catalog management routines.

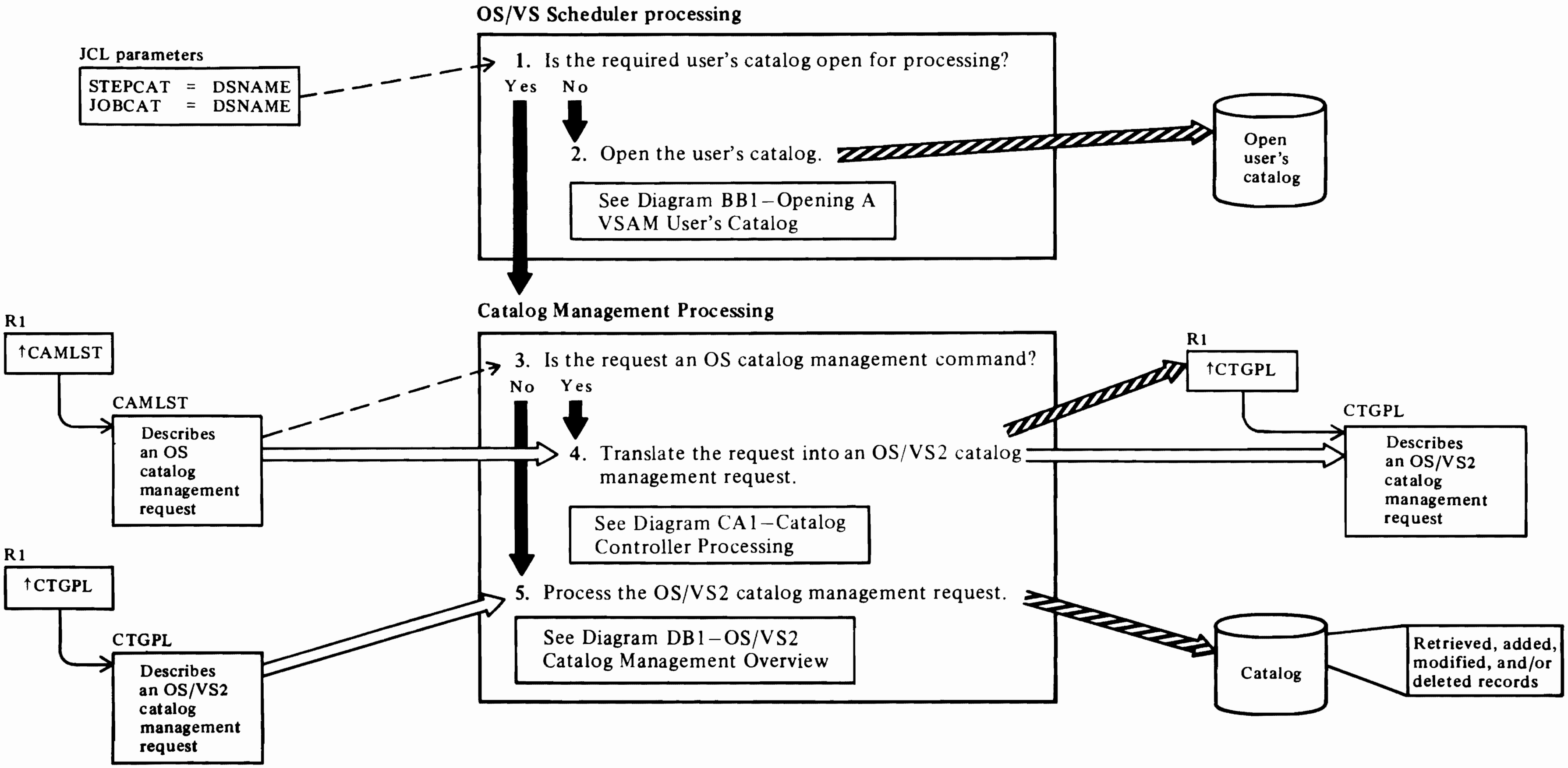
Figure 5. Notes to Method of Operation Diagram

The notes provide details about the processing shown in the diagram.

The notes also name the modules and routines that perform the functions represented. The module and procedure names allow you to relate a process step to a unit of code in the catalog management program listings.

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram AA1. Catalog Management Overview



Notes for Diagram AA1

Catalog management is called when information in one of the catalogs available to the caller is needed to open, extend (obtain more space for), or close a data set. Catalog management is also called when information must be added to, deleted from, or modified within one of the catalogs.

1 OS/VS Scheduler

2 IEFAB4F5

When the user's JCL DD JOBCAT or STEPCAT statements specify a user's (private) catalog, the OS/VS2 Scheduler calls IEFAB4F5 to open the user's catalog. See "Open Catalog Control Blocks," a figure in the "Data Areas" section that shows how the TCB, JSCB, and PCCBs are related to identify each open catalog available to the user's job.

3 IGC0002F

If the user has issued an OS catalog management request and has specified a CVOL volume serial number in the parameter list, then IGC0002F exits to the CVOL Processor, IGG0CLCA, to process the request in the specified CVOL (see *OS/VS2 CVOL Processor Logic*).

4 IGC0002F

When the user issues an OS catalog management command without a CVOL volser, the command is translated to a OS/VS2 catalog management request and is processed by OS/VS2 catalog management routines. When the processing completes, the return code (and other information for the user's program, as applicable) is translated to an OS catalog management return code.

Note: OS catalog management requests usually apply to NonVSAM, Alias, and GDG Base catalog records.

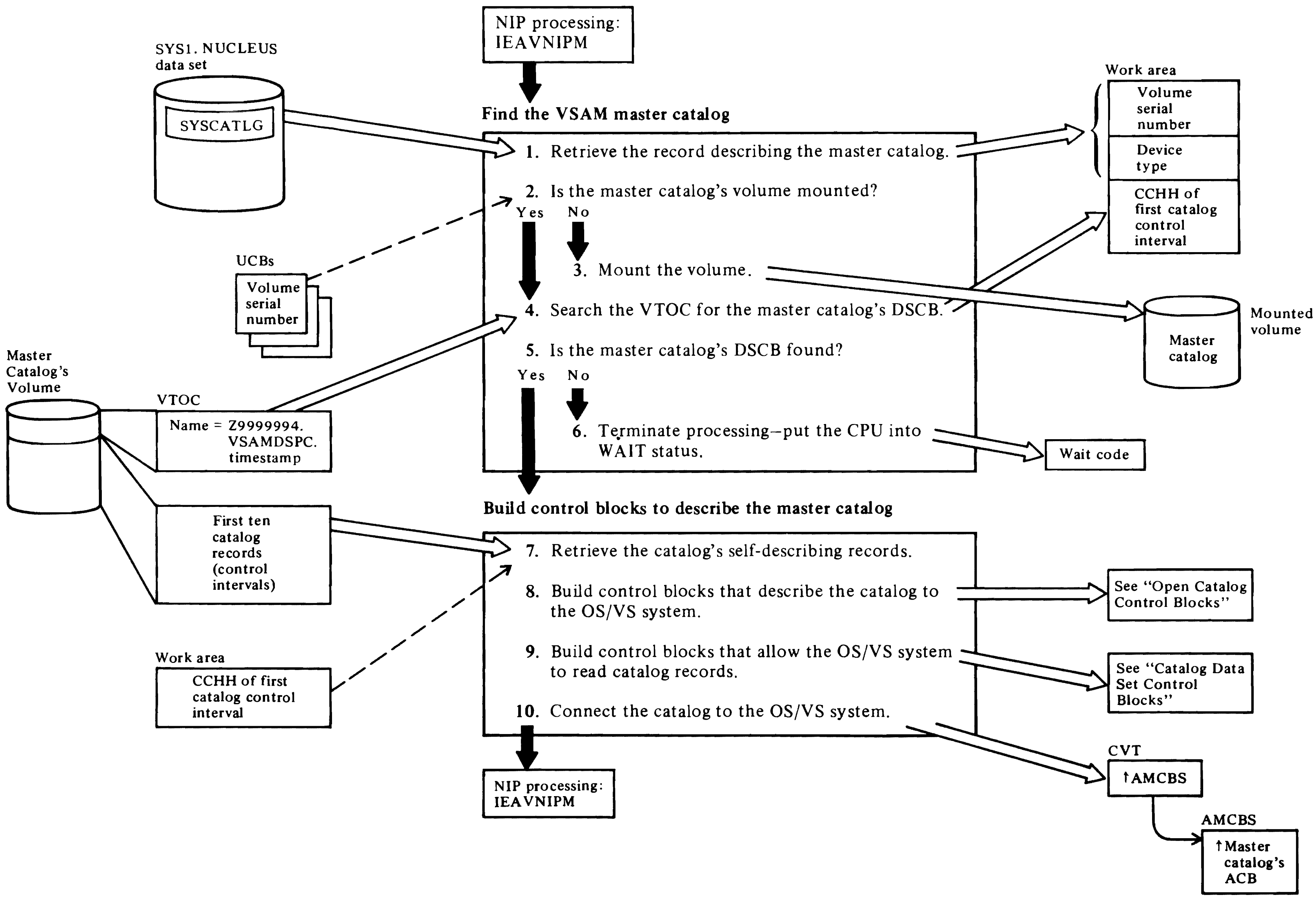
5 IGG0CLA1

When the user is an Access Method Services routine, a part of OS/VS2 catalog management, called "catalog management services", processes the request. Catalog management services routines are described in Method of Operations Diagrams EA1 through EN1.

When the user is a VSAM Open, Close, or End of Volume routine, OS/VS2 catalog management routines process the request. These routines are used to retrieve, add to, or update the information in catalog records. Catalog management services routines call the catalog management routines to manipulate catalog record information. Method of Operations Diagrams DA1 through DM3 describe the OS/VS2 catalog management routines.

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram BA1. Opening the Master Catalog



Notes for Diagram BA1

The master catalog is opened when the OS/VS2 system is started—during NIP (the nucleus initialization procedure). When NIP finishes, the master catalog's control blocks are freed (during a "psuedo-close') and the master catalog is opened a second time. The master catalog remains open as long as OS/VS2 is operational. The master catalog is never closed, even though it is conceptually described as a key-sequenced data set.

IEAVNIPM calls **IEAVNP11**

1 **IEAVNP11**

The SYS1.NUCLEUS data set contains a record, labeled "SYSCATLG", that describes the master catalog's location–its volume serial number.

2 **IEAVNP11**

Each mounted volume on the OS/VS system is described in one of the UCBs in the UCB table.

3 **IEAVNP11**

OS/VS issues a message to the operator to mount the volume containing the master catalog.

4 **IEAVNP11**

The volume's table of contents (VTOC) contains a DSCB identified by the name 'Z999999n.VSAMDSPC. Taaaaaaa.Tbbbbbbb', where 'n' is '4' or '6' and 'a' and 'b' are digits of the timestamp, that describes the space containing the first extent of the catalog's low-key range. This part of the catalog contains the catalog's self-describing catalog records in the first fourteen catalog control intervals.

5 **IEAVNP11**

6 **IEAVNP11**

7 **IEAVNP11**

8 **IEAVNP11**

The ACB, CAXWA, and AMCBS describe the master catalog to the OS/VS system.

9 **IEAVNP11**

The catalog is conceptually a key-sequenced VSAM data set divided into key ranges.

10 **IEAVNP11**

The address of the AMCBS is put into the CVT to connect the catalog control blocks to the OS/VS system control blocks.

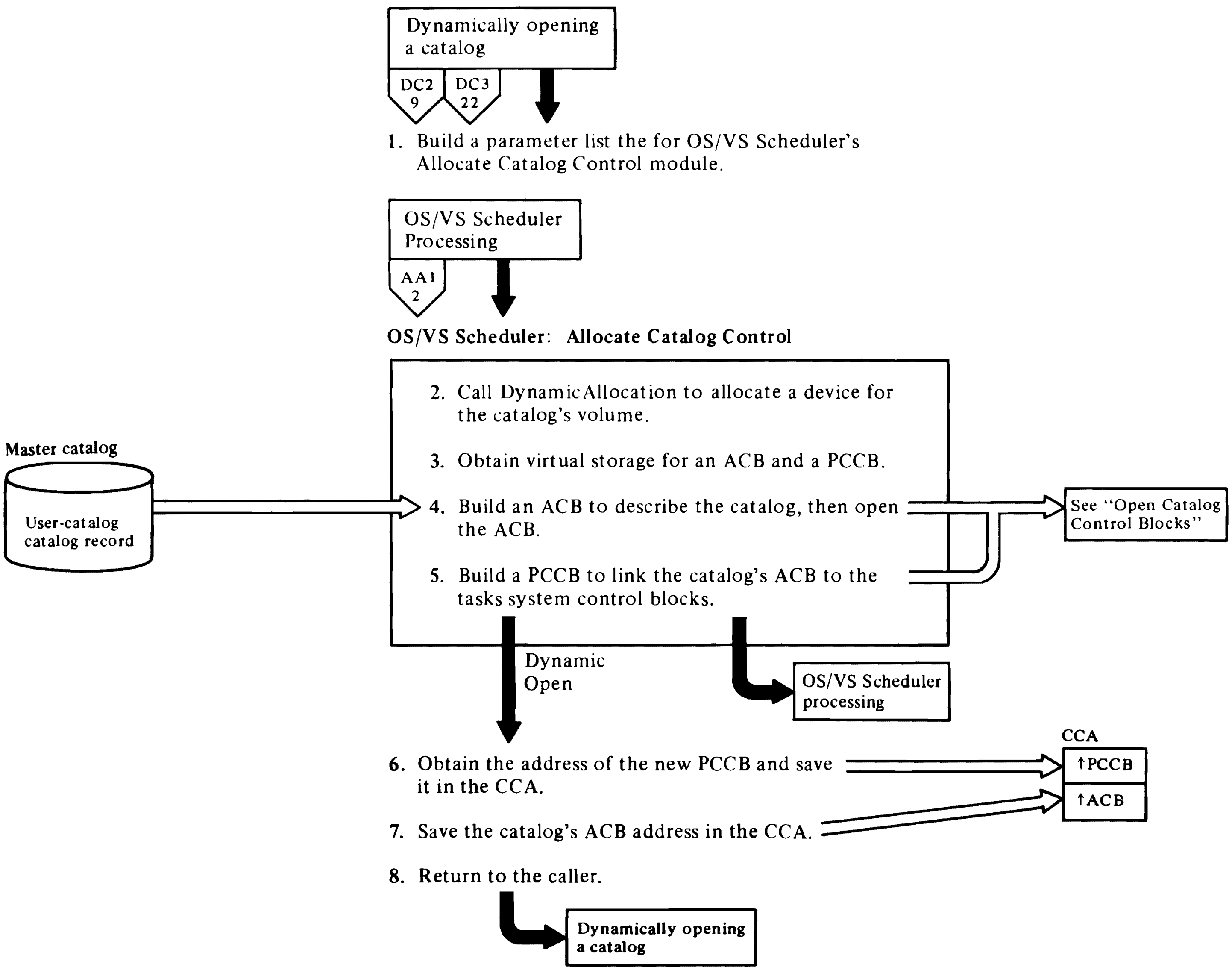
For additional information about topics related to opening the OS/VS2 master catalog, see:

"Data Areas:"

Control Block Interrelationship Figures
Open Catalog Control Blocks
VSAM Control Blocks That Describe a Catalog

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram BB1. Opening A User's Catalog



Notes for Diagram BB1

A user's catalog is opened when one of the following conditions occurs:

- When the catalog's DSNAME is specified in the user's JCL JOBCAT or STEPCAT DD statements, the OS/VS Scheduler calls IEFAB4F5 to open the catalog before the user's program begins processing.
- When an Access Method Services routine specifies a DSNAME that is in an unopened catalog, the catalog management search routine calls IEFAB4F5 to open the catalog. This process is called *dynamic opening* and occurs while the user's program is processing.

  The user's catalog remains open until all tasks that need to access the catalog have completed. When all tasks that use the user's catalog are complete, the OS/VS Scheduler closes the catalog.

  When the OS/VS Scheduler opens and closes a user's catalog, the process is similar in concept to a user's program opening and closing a key-sequenced data set.

1 IGG0CLA3: IGGPACI

2 IEFAB4F5

3 IEFAB4F5

The ACB and PCCB describe a user's catalog to the OS/VS system. The "Open Catalog Control Blocks" figure, in the "Data Areas" section, shows the control blocks that describe a catalog to the user program's TCB.

4 IEFAB4F5 calls **IDACAT11**

The ACB is opened by issuing an OPEN macro instruction, resulting in SVC 19, that refers to the ACB.

5 IEFAB4F5 calls **IEFAB4EF**

6 IGG0CLA3: IGGPGPCC

When a catalog is opened as a result of a catalog management request, the address of its ACB and PCCB is saved in the CCA.

7 IGG0CLA3: IGGPGPCC

8 IGG0CLA3: IGGPACI

For additional information about topics related to opening a user's catalog, see:

"Data Areas:"

Control Block Interrelationship Figures
Open Catalog Control Blocks
VSAM Control Blocks That Describe a Catalog

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram CA1. Catalog Controller Processing

R1: ↑CAMLST

CAMLST: Describes an OS catalog management request

R1: ↑CTGPL

CTGPL: Describes an OS/VS2 catalog management request

Workarea: Request type

CCAPROB: Return code

SVC 26

1. Obtain virtual storage for a workarea.

Request decoding

2. Is the request an OS catalog management command?
   No Yes
   - 2.1 Is there a CVOL volume serial in the CAMLST?
     No Yes
     - 2.2 Exit to CVOL Processor (IGG0CLCA) to process this OS catalog management request.
3. Translate the request into an OS/VS2 catalog management request.
4. Record the request type.

R1: ↑CTGPL

CTGPL: Describes an OS/VS catalog management request

Workarea: Request type

5. Process the OS/VS2 catalog management request.

See Diagram DB1–OS/VS2 Catalog Management Overview

Final processing after return from OS/VS2 catalog management

6. Was the request an OS catalog management command?
   No Yes
7. Translate a nonzero return code to the appropriate OS catalog management return code.
8. Free the virtual storage obtained for a workarea.

R15: Return code

9. Return to the caller.

CCAPROB: Return code

Notes for Diagram CA1

OS catalog management commands are used to process information in an OS system catalog. OS/VS2 catalog management can convert the OS commands and parameter lists to OS/VS2 catalog management commands and control blocks.

1 **IGC0002F**

(Receives control from dummy catalog module IGG026DU).

2 **IGC0002F: STRTCNTL**

When the caller issues a call to OS/VS2 catalog management, register 1 contains the address of a CTGPL—catalog parameter list.

When the caller issues an OS catalog management command, register 1 contains the address of a CAMLSTD control block—a parameter list provided by the caller of OS catalog management.

3 **IGC0002F: STRTCNTL**

If the OS catalog management request includes a CVOL volume serial number in the parameter list, then IGC0002F transfers control to the CVOL processor (IGG0CLCA), which processes the request in a CVOL and returns directly to the caller (not to IG0002F).

If no CVOL volume serial number is supplied, an OS catalog management request must be translated into an appropriate MVS catalog management request, as follows:

OS Request	MVS Request
CATALOG	DEFINE NONVSAM
UNCATALOG	DELETE NONVSAM
RECATALOG	ALTER NONVSAM
LOCATE by name	SUPERLOCATE
INDEX(BLDA, BLDG, BLDX, DLTA, DLTX, DRPX, LNKX)	DUMMY DELETE

A DUMMY DELETE request is a standard MVS DELETE request with a zero data set type in the CTGPL (CTGTYPE=0). It causes MVS catalog management to find the proper CVOL for this INDEX request based on the first qualifier of the data set name.

A LOCATE-by-block request without a CVOL volume serial specified will result in a return code of 4.

4 **IGC0002F: STRTCNTL**

Before the OS/VS request is issued, the original request's type (OS or OS/VS) is saved.

5 **IGC0002F** XCTLs to **IGG0CLA1**

6 **IGG0102F** (secondary entry point of IGC0002F)

7 **IGC0002F: RESMCNTL** calls **OSVOLST** and **OSRETCD**

8 **IGC0002F**

For additional information about topics related to catalog controller processing, see:

"Data Areas:"

Catalog parameter list (CTGPL) description and format

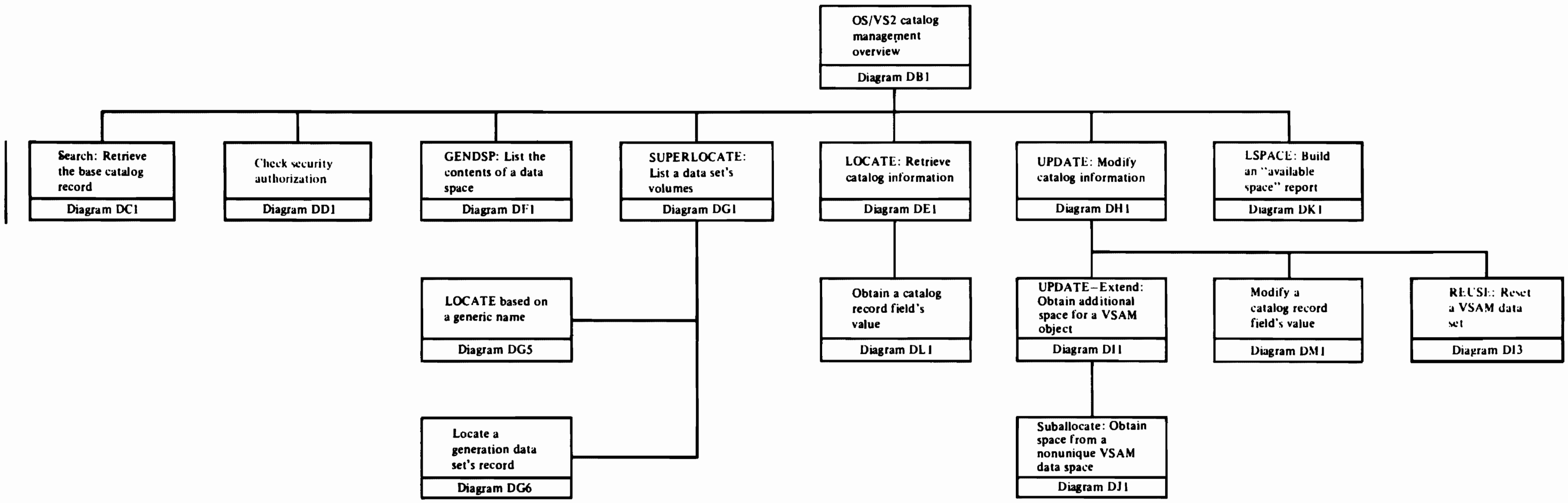
"Diagnostic Aids:"

Catalog management return codes

For additional information on dummy catalog module IGG026DU, see *OS/VS2 System Programming Library: Data Management*

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DA1. OS/VS2 Catalog Management Table of Contents

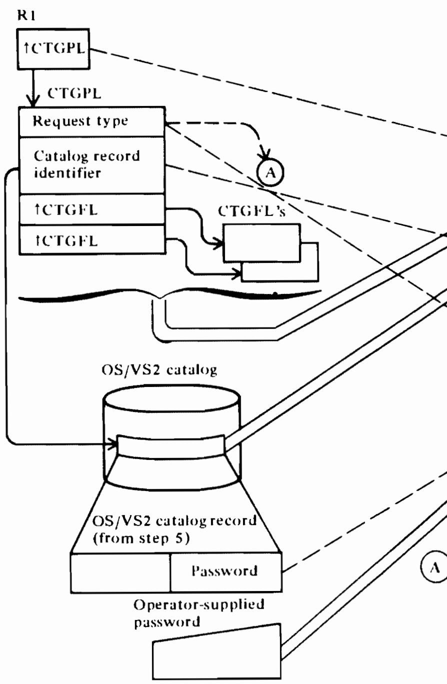


Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DB1. OS/VS2 Catalog Management Overview

VSAM Open, Close, and End-of-Volume, and Access Method Services issued CATLG (SVC 26) for an OS/VS2 Catalog Record. OS/VS2 Catalog Management begins here.



OS/VS2 catalog management

1. If caller is not in supervisor state or key 0, validity-check the CTGPL. If check fails, issue WTO message IEC338I and return to caller.
2. Issue GETMAIN for CCA. If GETMAIN fails, issue WTO message IEC339I and return to caller.
3. Establish a recovery environment.
4. Build and initialize a CCA for the caller's request.
5. Initialize the caller's CTGPL and CTGFLs.
6. Retrieve the catalog record identified by the caller: (See Diagram DC1, "Search: Retrieve the Base Catalog Record.")
7. If the caller issued an Access Method Services command, process it and bypass steps 3 through 5. (See Diagram EB1, "Catalog Management Services Overview.")
8. Check the caller's authorization to use the record: (See Diagram DD1, "Check Security Authorization.")

   Authorized — Not authorized → 10

9. (A) Determine the request type and process it:
   - SUPER-LOCATE → (See Diagram DG1, "SUPERLOCATE, Build a List of all Volumes Associated with a DSNAME.")
   - GENDSP → (See Diagram DF1, "GENDSP: Build a List of all Data Sets in a VSAM Data Space.")
   - LOCATE → (See Diagram DE1, "LOCATE: Retrieve Catalog Information.")
   - UPDATE → (See Diagram DH1, "UPDATE. Modify Catalog Information.")
   - LSPACE → (See Diagram DK1, "LSPACE: Build an 'Available Space' Report.")

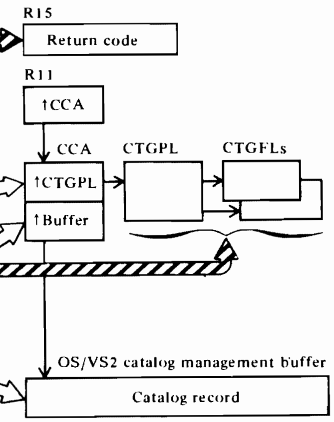
EB3 22 — EB2 14,19

10. Release virtual storage obtained for this request.
11. Write problem determination message and set error code, if any.
12. Release recovery environment.

OS/VS catalog management

CCACD1 Return code

13. Return to the caller. → Register 15 Return code



Notes for Diagram DB1

Catalog Management is called when VSAM Open, Close, End of Volume, or the Access Method Services routines issue the CATLG macro instruction (SVC 26). Register 1 contains the address of the caller's catalog parameter list. The catalog parameter list identifies which catalog record to process and what process to perform.

Catalog Management is also called when the user's program issues OS Catalog Management macro instructions: CAT, LOC, RECAT, and UNCAT. The user's program builds a CAMLST parameter list to describe the request. Register 1 points to the CAMLST.

A user's program can access the OS/VS2 catalog by issuing an Access Method Services utility request. Access Method Services translates the request into an SVC 26 and a catalog parameter list.

IGC0002F

Receives control from dummy catalog module IGG026DU.

Register 1 contains the address of a catalog parameter list (CTGPL). OS/VS Catalog Management transfers control (XCTL) to OS/VS2 catalog management load module, IGG0CLA1.

IGG0CLA1 calls IGG0CLC9 to process the VSAM catalog management request.

1 **IGG0CLC9: BLDCCA**

A call is made to the task supervisor validity-check routine to verify that the storage passed as a CTGPL is owned by the caller. A condition code of 8 is set in the PSW if the check is successful.

2 **IGG0CLC9: BLDCCA**

Issue a page boundary GETMAIN for CCA and record areas. If return code is not zero, issue message. Set return code 8 in register 15 if caller was a SUPERLOCATE request or a translated request. If it was not, set reason and error code and module ID in the CTGPL.

3 **IGG0CLC9: BLDCCA**

Issue ESTAE to establish recovery environment.

4 **IGG0CLC9: BLDCCA**

The catalog control area (CCA) contains data about catalog records retrieved to process the request. The CCA also contains a register save area that shows the flow of control between catalog management routines used to process the request.

Each time a catalog management routine calls another catalog management routine, the contents of registers 12, 13, and 14 are put in the CCA's register save area. Register 13 contains the address of the next 12-byte register save area in the CCA. Register 12 contains the address of the calling routine. Register 14 contains the return address to the calling routine.

5 **IGG0CLAB: IGGPACDV** (calls **IGGPSCNC (IGG0CLAY)**)

The caller's work area and each CTGFL are checked to ensure that it is within the caller's address space.

The CTGFL's field-name value is used to obtain dictionary data that defines the field's characteristics and location within the record.

6 **IGG0CLAB: IGGPACDV** (calls **IGGPSCAT (IGG0CLAH)**)

The catalog record is identified by the caller's dsname value, volume serial number, or control interval number.

7 **IGG0CLAB: IGGPACDV** (calls **IGGPCDVR (IGG0CLAT)**)

An Access Method Services command is translated into a catalog management services request to create, modify, delete, or list catalog records.

8 **IGG0CLAB: IGGPACDV** (calls **IGGPCKAU (IGG0CLBM)**)

The caller's request type determines the level of authorization that allows the OS/VS2 catalog management routines to complete the caller's request.

9 **IGG0CLAB: IGGPACDV** (calls **IGGPSLOC (IGG0CLAM), IGGPGDSP (IGG0CLBJ), IGGPLOC (IGG0CLAZ), IGGPUPD (IGG0CLAV),** or **IGGPLSP (IGG0CLBK)**)

IGGPSLOC:

A SUPERLOCATE request builds a list of all volumes and units associated with a dsname.

IGGPGDSP:

A GENDSP request builds a list of all VSAM data sets in a VSAM data space.

IGGPLOC:

A LOCATE request retrieves information from the catalog record.

IGGPUPD:

An UPDATE request modifies information in a catalog record. An UPDATE request can also obtain direct-access space for the data set or index identified by the dsname value.

IGGPLSP:

A LSPACE request determines the amount of available space on a direct-access volume, when the volume is described in a VSAM catalog.

10 **IGG0CLC9: IGGPRCU**

When the OS/VS2 catalog management request is complete, all virtual storage obtained for work areas, control blocks, and the request's CCA is returned to the OS/VS system.

11 **IGG0CLC9: IGGPRCU**

Write problem determination message if it was a SUPERLOCATE request or a translated request and set error code in CTGPL.

12 **IGG0CLC9: IGGPRCU**

Issue ESTAE to remove recovery environment.

13 OS/VS2 catalog management common processing (IGG0CLC9) sets a return code in register 2 and returns to IGG0CLA1.

IGG0CLA1 transfers control (via XCTL) to IGG0102F, an entry point in IGC0002F. If OS/VS2 catalog management determines that CVOL processing is required, IGG0CLA1 exits to IGG0CLCA—the OS/VS CVOL processor.

IGC0002F puts the return code into register 15 and returns to the caller via the SVC return.

For additional information about topics related to the OS/VS2 catalog management overview, see:

"Data Areas:"

Catalog communications area (CCA) format and description
Catalog parameter list (CTGPL) format and description
Field name dictionary

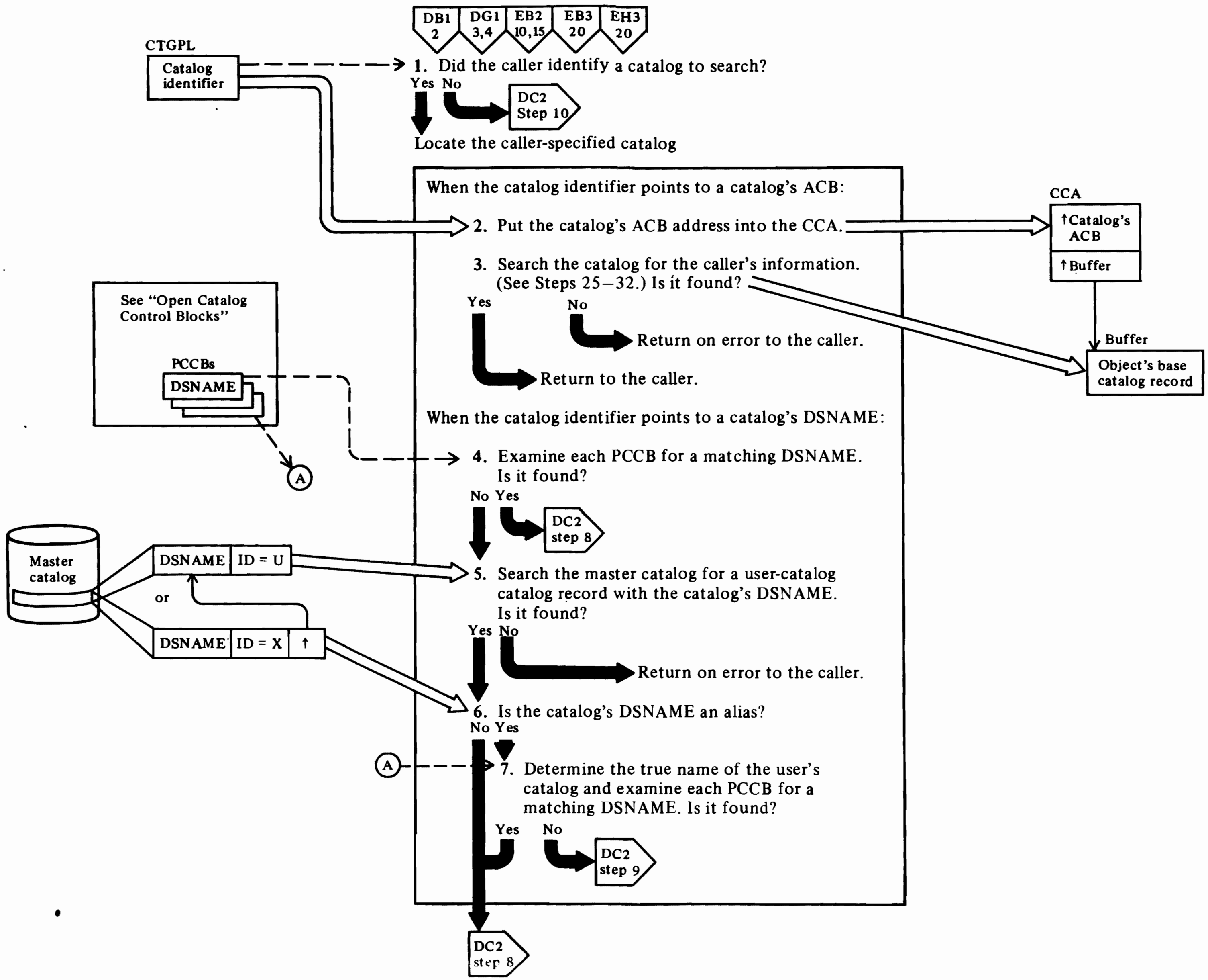
"Diagnostic Aids:"

CCA register save area
Catalog management return codes

For additional information on dummy catalog module, see *OS/VS2 System Programming Library: Data Management*

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DC1. SEARCH: Retrieve the Base Catalog Record, Search the Caller-Specified Catalog



Notes for Diagram DC1

1 **IGG0CLAH: IGGPSCAT** calls **IGGPSC2 (IGG0CLAH)**

The CTGPL's catalog identifier field, set by the caller, can contain the address of a catalog's ACB, the address of a catalog's DSNAME, or 0.

2 **IGG0CLAH: IGGPSC2**

The catalog specified by the caller is the only catalog searched. The Catalog Management Services DEFINE routine calls the Search routine to confirm that, when a caller wants to create a VSAM cluster or catalog, the new cluster or catalog DSNAME isn't duplicated in the catalog. The caller (Catalog Management Services DEFINE routine) expects the "no record found" return code.

3 **IGG0CLAH: IGGPSC2** calls **IGGPDOG (IGG0CLA3)**

4 **IGG0CLAH: IGGPPCCP** calls **IGGPGPCC (IGG0CLA3)**

If the CTGPL's catalog identifier field contains the address of a catalog DSNAME, the search routine examines each protected catalog control block (PCCB) for a matching DSNAME field. Each PCCB contains the address of its catalog's ACB.

If no PCCB contains a matching DSNAME, the user-supplied catalog DSNAME refers to either a nonexistent catalog or to an unopened catalog.

5 **IGG0CLAH: IGGPPCCP** calls **IGGPDOG (IGG0CLA3)**

The AMCBS (addressed by the CVT) contains the address of the VSAM master catalog's ACB.

6 **IGG0CLAH: IGGPPCCP**

7 **IGG0CLA3: IGGPDOG** calls **IGGPGRE**

IGG0CLAH: IGGPPCCP calls **IGGPGPCC (IGG0CLA3)**

For additional information about topics related to Search processing, see:

"Data Areas:"

Cluster catalog record description
Volume catalog record description
Access method control block (ACB) format and description
Catalog communications area (CCA) format and description
Catalog parameter list (CTGPL) format and description
CTGPL search options
Private catalog control block (PCCB) format and description
Access method control block structure (AMCBS) format and description

Open Catalog Control Blocks

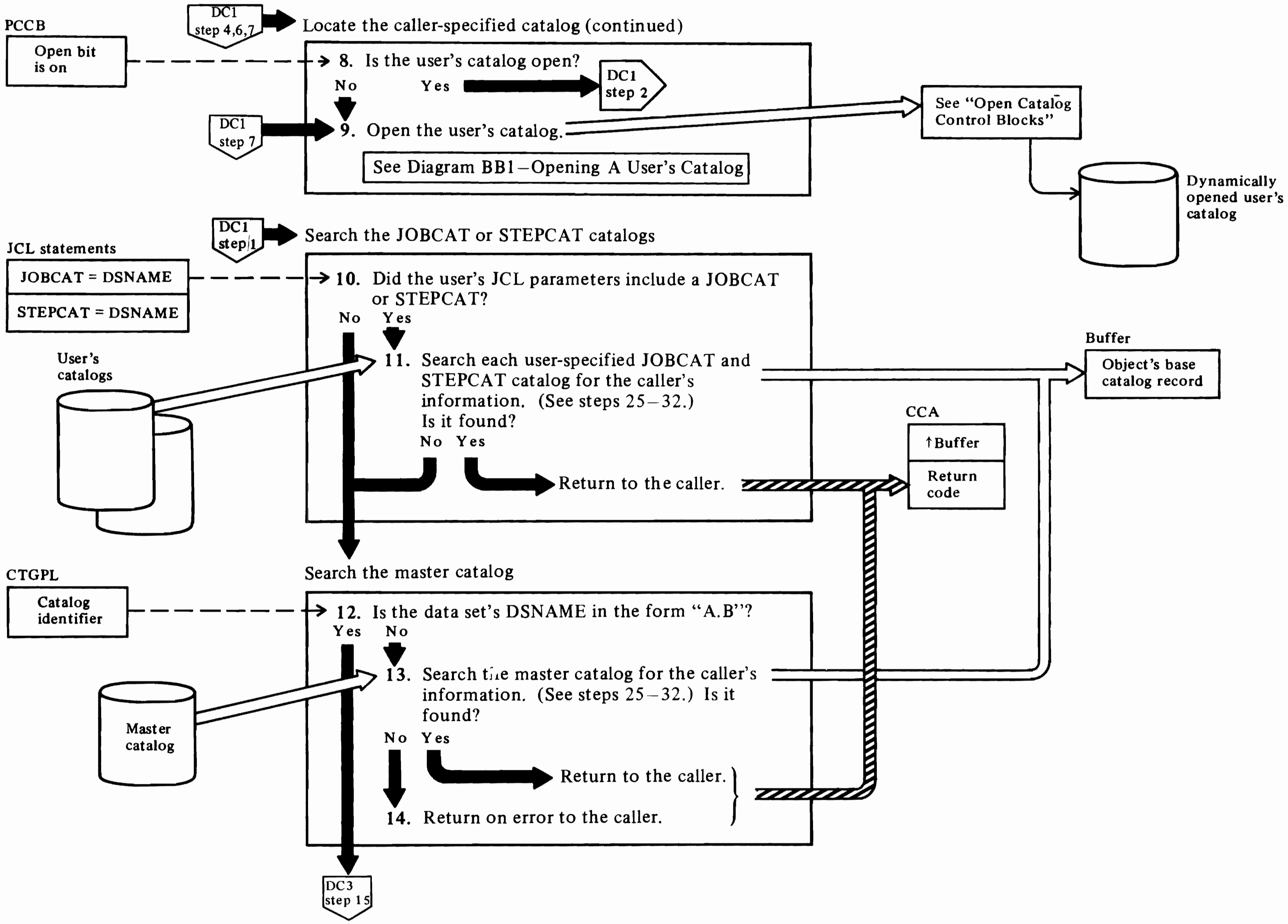
"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DC2. SEARCH: Retrieve the Base Catalog Record, Search All Catalogs Available to the Caller



Notes for Diagram DC2

8 IGG0CLAH: IGGPPCCP

9 IGG0CLAH: IGGPPCCP calls **IGGPACI**

10 IGG0CLAH: IGGPSC2 calls **IGGPCSCP (IGG0CLA3)**

If the CTGPL's catalog identifier field contains 0, the user catalogs specified by the user's JCL JOBCAT and STEPCAT DD statements, and the master catalog, are searched until either the record is found or there are no more catalogs to search.

The JSCB contains the address of the first PCCB in the PCCB chain. Each PCCB describes one of the user catalogs that have been opened to satisfy the user's JCL JOBCAT and STEPCAT DD statements. A PCCB contains the address of a catalog's ACB. The catalog's ACB address is put in the CCA to identify the catalog being searched.

11 IGG0CLA3: IGGPCSCP calls **IGGPDOG**

12 IGG0CLA3: IGGPCSCP

13 IGG0CLA3: IGGPCSCP calls **IGGPDOG**

14 IGG0CLA3: IGGPCSCP

For additional information about topics related to Search processing, see:

"Data Areas:"

Cluster catalog record description
Volume catalog record description
Access method control block (ACB) format and description
Catalog communications area (CCA) format and description
Catalog parameter list (CTGPL) format and description
CTGPL search options
Private catalog control block (PCCB) format and description
Access method control block structure (AMCBS) format and description

Open Catalog Control Blocks

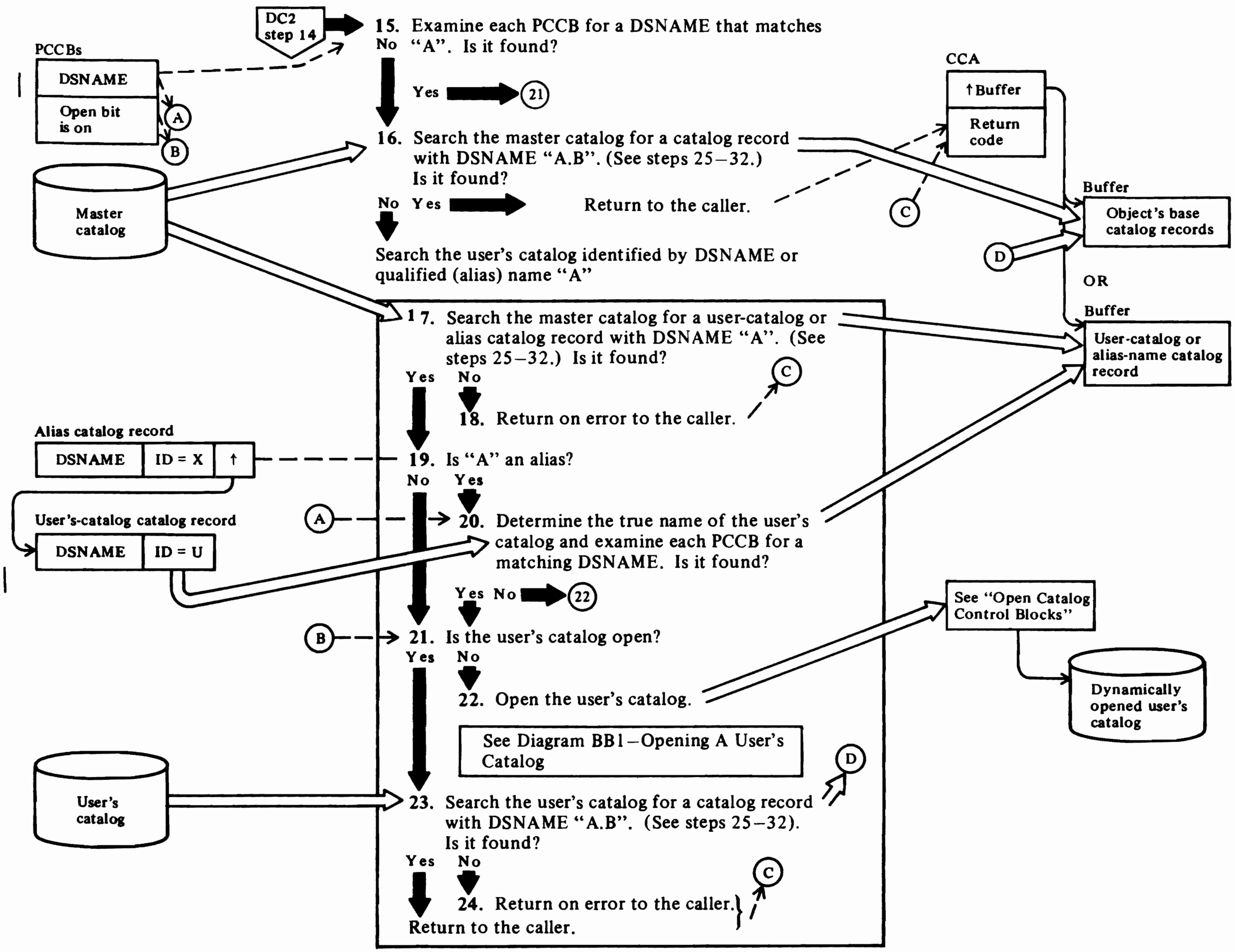
"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DC3. SEARCH: Retrieve the Base Catalog Record, Search the Catalog Identified by a Qualified Name

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912



Notes for Diagram DC3

15 IGG0CLA3: IGGPCSCP calls **IGGPCPCC**

16 IGG0CLA3: IGGPCSCP calls **IGGPDOG**

17 IGG0CLA3: IGGPCSCP calls **IGGPDOG**

18 IGG0CLA3: IGGPCSCP

19 IGG0CLA3: IGGPDOG

20 IGG0CLA3: IGGPDOG calls **IGGPGRE**

If the PCCB identifies a CVOL-type catalog, IGGPDOG initializes the controller's work area for CVOL processing and exits to IGG0CLCA—the OS/VS CVOL processor.

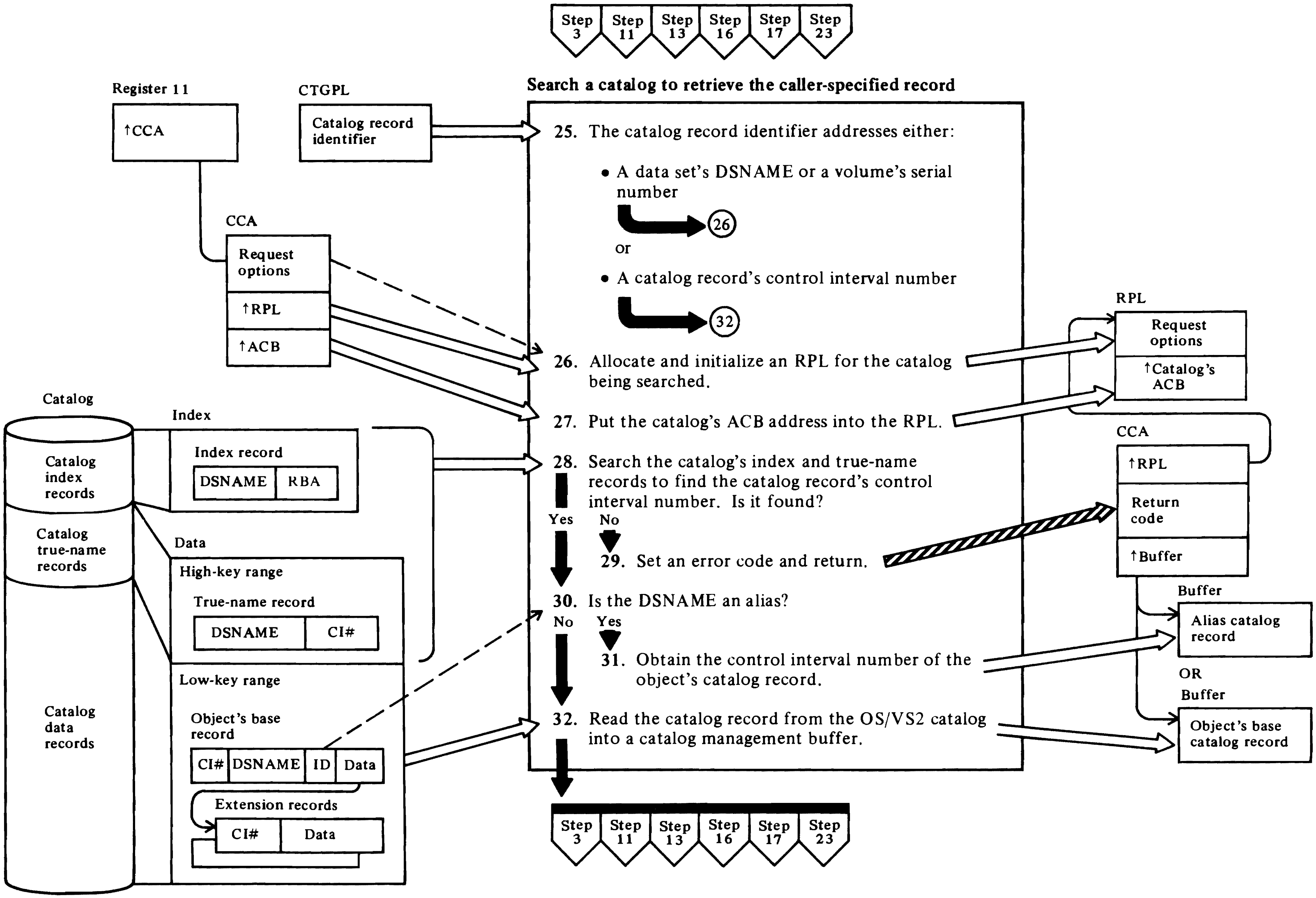
21 IGG0CLA3: IGGPCSCP

22 IGG0CLA3: IGGPCSCP calls **IGGPCSDO (IGG0CLAH)**

23 IGG0CLA3: IGGPDOG

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DC4. SEARCH: Retrieve the Base Catalog Record, Retrieve the Caller-Specified Record



Notes for Diagram DC4

25 IGG0CLA3: IGGPDOG

26 IGG0CLA3: IGGPRPLF and **IGGPRPLM**

The search routine assigns one request parameter list (RPL) to the caller. Catalog management routines issue GET and PUT macro instructions to retrieve and write catalog records. Each record-management request (GET, PUT, etc.) needed to satisfy the caller's catalog-management request refers to this RPL. This RPL is initialized for a caller and used as often as necessary to process the caller's catalog-management request. When the caller's catalog-management request is completed, the RPL is assigned to another caller.

27 IGG0CLA3: IGGPGPCC

28 IGG0CLA3: IGGPDOG

The goal of the search is to find the true-name record identified by the DSNAME or the volume serial number. The true-name record contains the DSNAME or volume serial number and the control interval number of the cluster, data, index, nonVSAM, alias, GDG base, or volume catalog record.

29 IGG0CLA3: IGGPDOG

30 IGG0CLA3: IGGPDOG

31 IGG0CLA3: IGGPDOG calls **IGGPGRE**

32 IGG0CLA3: IGGPDOG

The catalog record is located by its control interval number and read into a catalog management buffer. The buffer's address is put into the CCA.

For additional information about topics related to Search processing, see:

"Data Areas:"

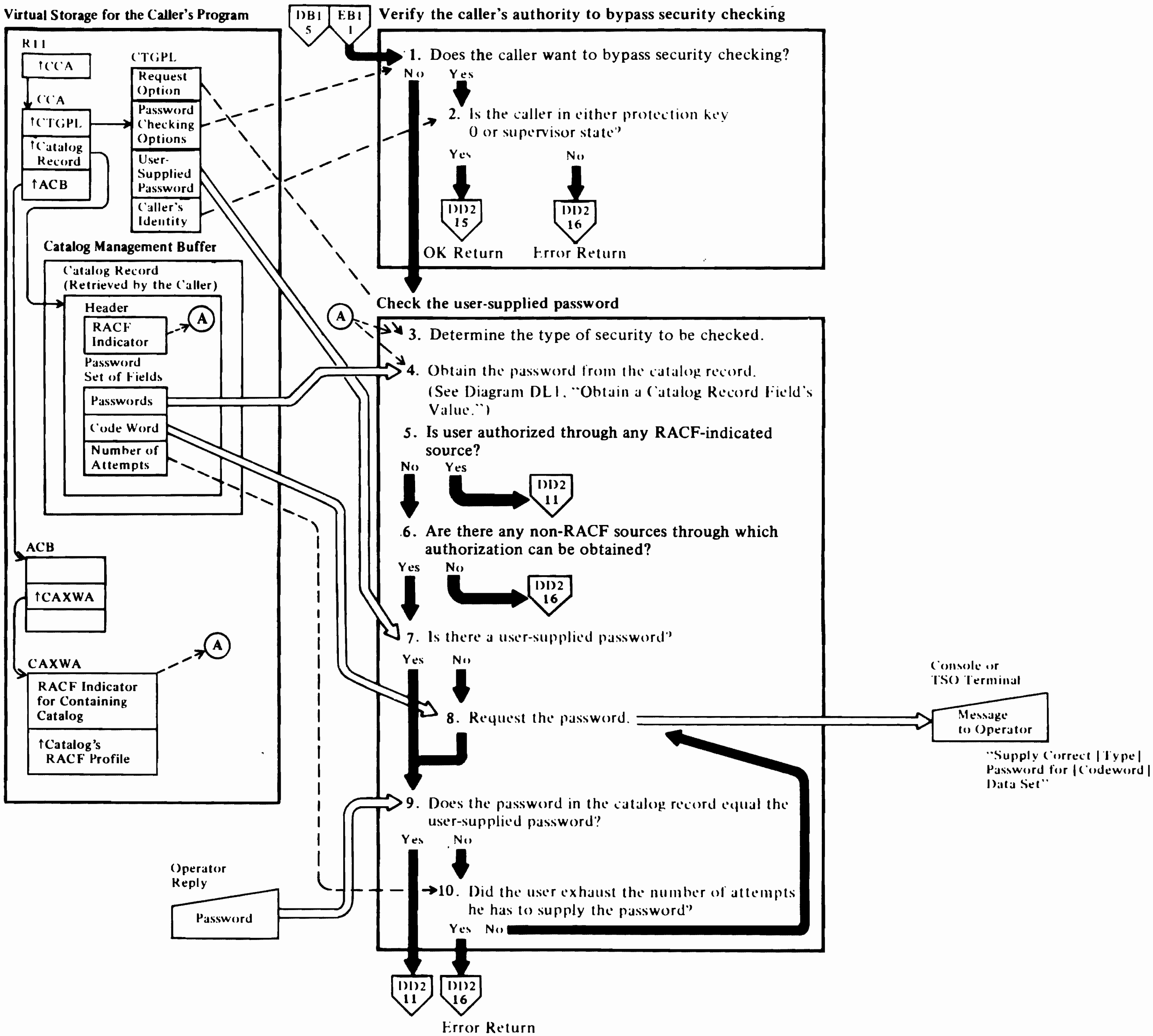
Catalog record general format and description
Catalog communications area (CCA) format and description

"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DD1. Check Security Authorization



Notes on Diagram DD1

IDA0192C and **IGG0CLAB: IGGPACDV** (calls **IGGPCKAU (IGG0CLBM))**

When the VSAM Open routine (IDA0192C) calls OS/VS2 Catalog Management to retrieve a cluster catalog record, the security checking routine confirms the user's authorization to gain access to the cluster. See VSAM Open Processing, in *OS/VS2 Virtual Storage Access Method (VSAM) Logic*, for details about connecting a data set to a user's program (open processing).

IGG0CLAT: IGGPCDVR (calls **IGGPCKAU (IGG0CLBM))**

When an Access Method Services routine calls a catalog management services routine, the security checking routine confirms the user's authorization to gain access to the catalog or to a specific catalog record.

The catalog record containing the password(s) is available in the buffer addressed by the caller's CCA.

The RACF indicator is contained in the catalog record for VSAM data sets. If an entity is protected by RACF and passwords, only RACF authorization will be checked. In some cases, the source of authorization to an item may be obtained either from the item itself or from another source. For example, one can export a cluster by having an authorization to the cluster or to the catalog that contains the cluster.

The type of processing that the user is allowed to do with the data set is determined by the password or RACF authorization. The levels of authorization are:

- RACF alter authorization or master password: The user is allowed to modify passwords and catalog records that describe his data set, and to process his data set's control intervals and records.
- RACF control authorization or control-interval password: The user is allowed to process the data set's control intervals as wells as its records.
- RACF update authorization or update password: The user is allowed to process his data set's records.
- RACF read authorization or read-only password: The user is allowed to read, but not to write (add or update), records in his data set.

1 IGG0CLBM: IGGPCKAU

If the user's security has been verified during a previous catalog management request, the caller (VSAM Open) can set the CTGPL's bypass-security-checking flag on.

2 IGG0CLBM: IGGPCKAU

Other catalog management callers, such as the user's program (with Access Method Services commands), and utility programs, are not in protection key 0 or supervisor state. If these programs attempt to bypass security checking, the security checking routine sets an error return code that prevents further catalog management processing for the caller's program.

3 IGG0CLB6: IGGPSPSC

Check for special situations and determine what levels of security need to be verified and on which entity security verification is to occur. Check, also, for cases in which no verification is required.

4 IGG0CLBM: IGGPCKEX

The password is in the password set of fields in the cluster, data, or index catalog record. The CTGPL can contain a password that the user supplied in a JCL statement.

5 IGG0CLBM: IGGPPWVR (calls IGGPRACV (IGG0CLDC))

If a source through which one may obtain authorization is RACF-protected, IGGPRACV is called to check for RACF authorization. If RACF authorization to the entity cannot be obtained through the RACF sources, then password checking is performed on the non-RACF sources.

8 IGG0CLBM: IGGPPWGT

The console operator, or TSO user, can reply to the VSAM request-for-password message with a password.

9 IGG0CLBM: IGGPPWVR

10 IGG0CLBM: IGGPPWVR

For additional information about topics related to password checking processing, see:

"Data Areas:"

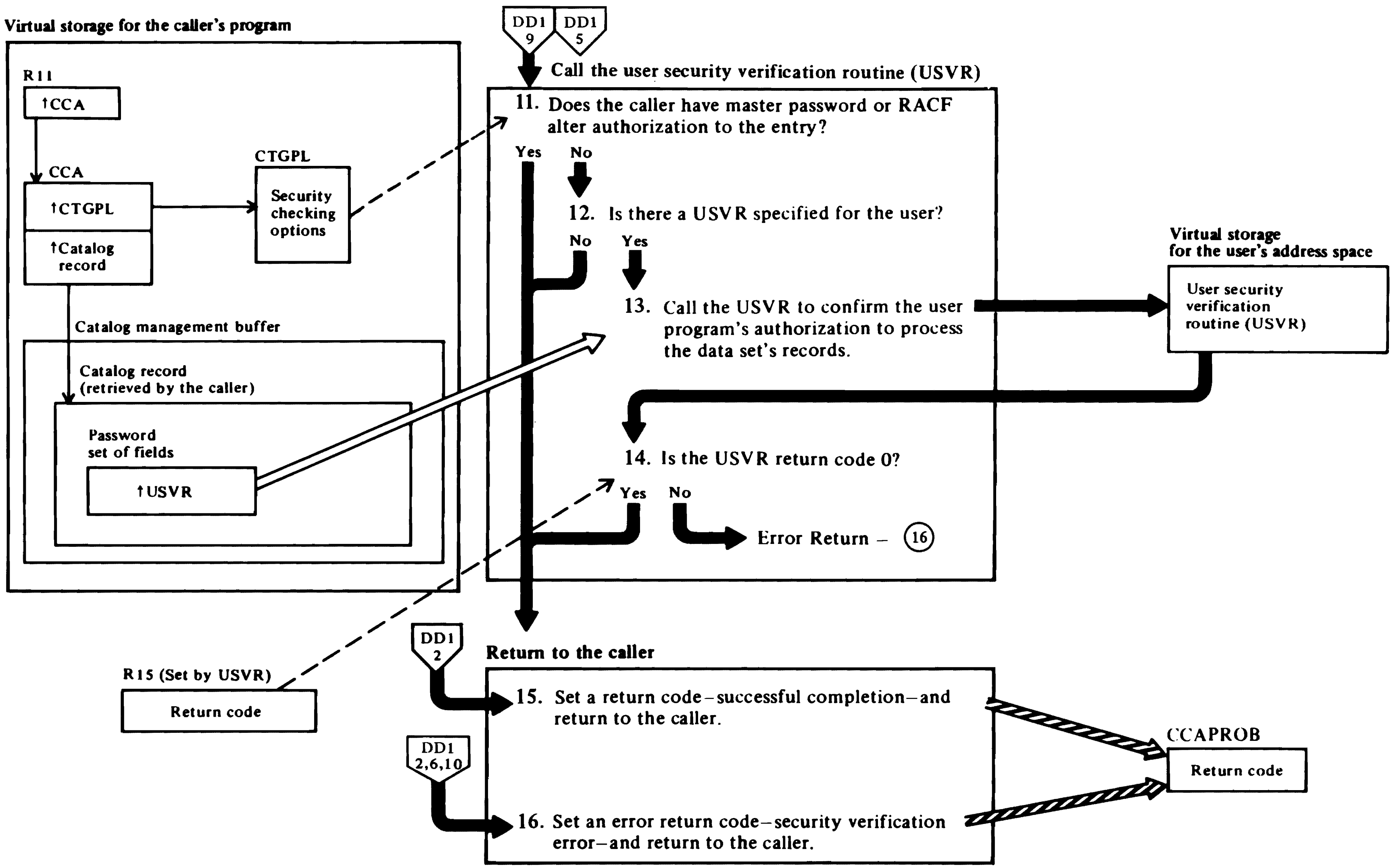
Password set of fields description

"Introduction:"

Passwords
Password processing restrictions

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DD2. Check Security Authorization



Notes for Diagram DD2

11 **IGG0CLB6: IGGPINMD**

If the user supplied the correct master password or has RACF alter authorization, the user security verification routine (USVR), if it exists, is bypassed. If a USVR exists, the USVR exit is taken only if the user provided another type of password correctly.

12 **IGG0CLB6: IGGPINMD**

If a user security verification routine exists, its name is in the catalog record's password set of fields.

13 **IGG0CLB6: IGGPINMD**

The user security verification routine (USVR) is an installation-supplied routine that confirms a user's authorization to gain access to the data set. The USVR confirms that the user satisfies the installation's security verification criteria.

For additional information about topics related to security checking processing, see:

"Data Areas:"

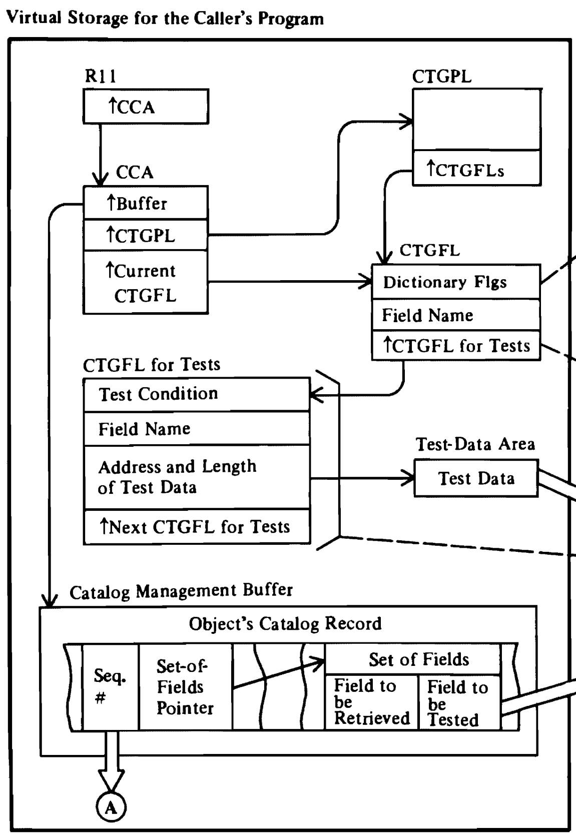
Cluster catalog record description
Password set of fields description

"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DE1. LOCATE: Retrieve Catalog Information



DB1 9 | DI3 4

1. Obtain information about each CTGFL's field name.

2. Do steps 3 through 16 for each of the caller's CTGFLs.

3. Is the field to be retrieved an upgrade field name?

No Yes

4. Use the associations in the current entry to read the upgrade entry into the catalog management buffer. (B)

5. Is the field to be retrieved only if caller-specified test conditions are satisfied?

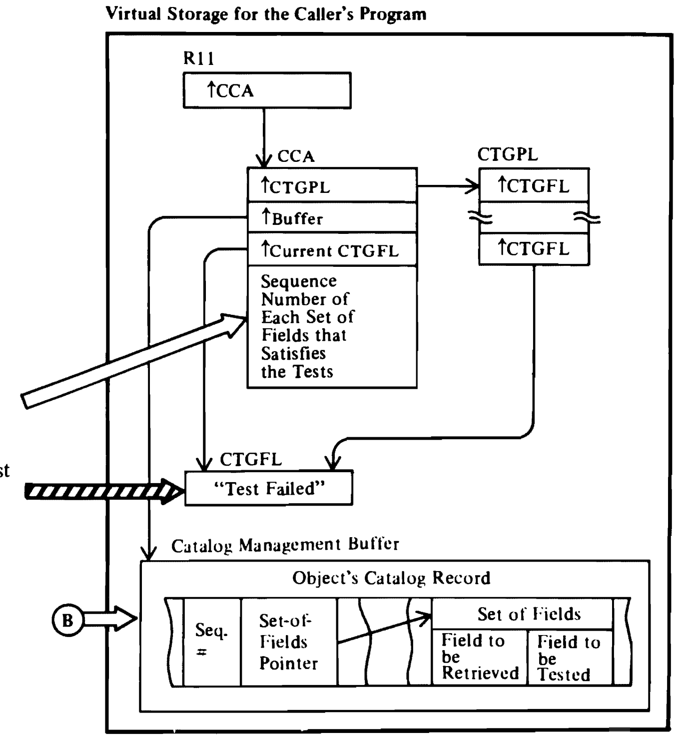
No Yes

6. Identify each set of fields that satisfies the test conditions.

7. Does at least one set of fields satisfy the test conditions.

Yes No

DE2 8 | DE2 18



Notes for Diagram DE1

IDA0192C

The VSAM Open routine issues the CATLG macro instruction (SVC 26) to obtain data set and volume information about the user's data set and index. See VSAM Open Processing, in *OS/VS2 VSAM Logic*, for details.

IDA0557A

The VSAM end of volume routine issues the CATLG macro instruction (SVC 26) to obtain volume information about the extents added to the user's data set. See VSAM End-of-Volume Processing, in *OS/VS2 VSAM Logic*, for details.

IGG0CLAB: IGGPACDV (calls **IGGPLOC (IGG0CLAZ)**)

When the caller issues a CATLG macro instruction, register 1 points to the caller's catalog parameter list (CTGPL). The CTGPL's request options are decoded and the base catalog record is retrieved for the request. See Diagram DB1, OS/VS2 Catalog Management Overview, for details about initial catalog management processing and request decoding.

IGG0CLB7: IGGPRUS, IGGPFRWK (calls **IGGPLOC (IGG0CLAZ)**)

Upon completion of Reuse processing, LOCATE is called to return catalog field information from the reset entry.

1 **IGG0CLAZ: IGGPEXT** (calls **IGGPSCNC (IGG0CLAY)**)

Each CTGFL is initialized with the dictionary entry associated with the CTGFL's field-name value. Calls from within catalog management (as opposed to external calls, such as LOCATE) enter at this point to use the field management retrieval function.

2 **IGG0CLAZ: IGGPSCNF**

Steps **3** through **16** are performed for each of the caller's CTGFLs.

The Locate routine processes each CTGFL associated with the caller's CTGPL and returns as much caller-requested data (in the caller's work area) as the caller's test conditions and work area size permit.

3 **IGG0CLAZ: IGGPSCNF** (calls **IGGPUPGD)**

A caller may request catalog information from an associated upgrade entry by using upgrade field names.

4 **IGG0CLAZ: IGGPUPGD**

The upgrade entry may not be in the catalog management buffer. If it is not in the buffer, the associations in the current entry are used to retrieve the upgrade entry.

5 **IGG0CLAZ: IGGPSCNF**

The caller's CTGFL list contains the address of each CTGFL required to satisfy the caller's need for catalog information. Each CTGFL describes one of the catalog record fields to be retrieved. Each CTGFL is completely processed before the next one is started.

IGG0CLAZ: IGGPSCNF (calls **IGGPTSTS (IGG0CLBA)**)

A caller might make conditional requests for retrieval of catalog record fields. For example, a chain of CTGFLs might be supplied with the request and processed together. The first CTGFL identifies a field to be retrieved and points to subsequent CTGFLs that contain the names of catalog fields to be tested, the test conditions (equal, low, high, etc.), and the address and length of the caller's test data area. The catalog record fields identified by the second and subsequent CTGFLs are compared to (or tested against) the caller's data. If the comparison satisfies the test conditions, the catalog record field specified by the first CTGFL is retrieved.

6 **IGG0CLBA: IGGPTSTS**

If the caller wants to retrieve a catalog record's header field, the field's data is retrieved if all tests are satisfied.

If the caller wants to retrieve a field from one of the sets of fields that follow the header fields, the field's data is retrieved from each set of fields that satisfies all tests.

7

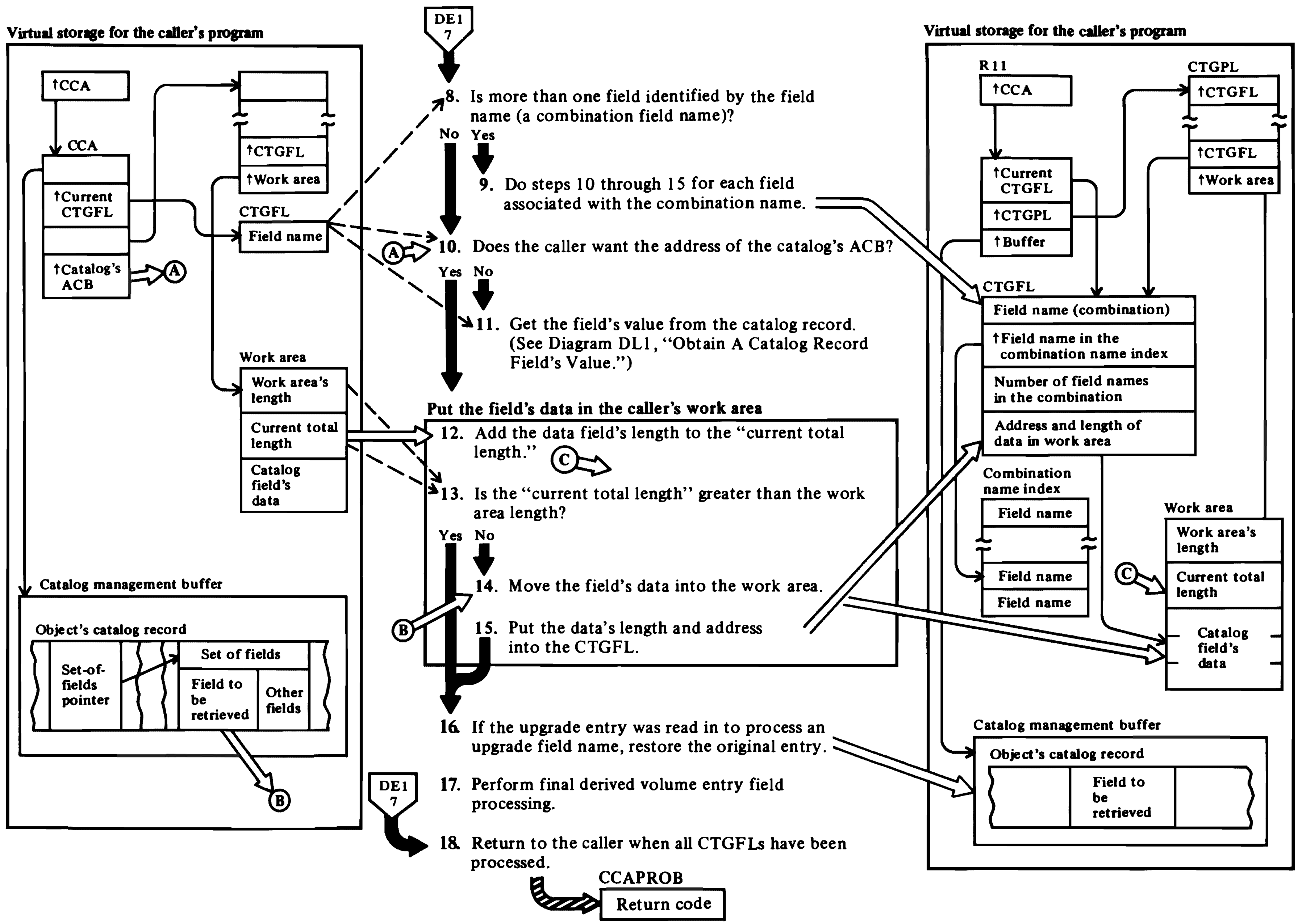
The sequence number of each set of fields that satisfies the tests is put in the CCA. After the sets of fields have been tested, the sequence numbers in the CCA are used to identify each set of fields that contain caller-requested data.

For additional information about topics related to LOCATE processing, see:

"Data Areas:"

Catalog record description and format

Diagram DE2. LOCATE: Retrieve Catalog Information



Notes for Diagram DE2

8 IGG0CLAZ: IGGPLOC2

A combination name refers to a set of related catalog field names, and is used by the caller instead of a separate CTGFL for each field name.

9 IGG0CLAZ: IGGPLOC2

The combination name index has an entry for each field name in the combination. The Locate routine processes each field name entry in the combination name index sequentially, starting at the index of the first field name entry for the combination, and ending when the number of entries processed equals the number of field names associated with the combination name.

The test sequence (if any) associated with a combination-name CTGFL is done only once, not once for each field name in the combination.

10 IGG0CLAZ: IGGPLOC2

The address of the catalog's ACB is in the CCA. All other catalog record fields that the caller can request are in the catalog record. Each catalog record field is identified by its field name.

11 IGG0CLAZ: IGGPLOC2 (calls **IGGPGVAL (IGG0CLBA)**)

Diagram DL1, Obtain A Catalog Record Field's Value, shows how the requested catalog record field (specified by its field name in the CTGFL) is located for the Locate routine.

12 IGG0CLAZ: IGGPSHIN

The first two fields in the caller's work area specify the number of bytes the caller allocated to the work area and the number of bytes that contain catalog record field data (the "current total length" field). If the "current total length" exceeds the work area length, the current total length field is updated with the length of the catalog record data, but the data itself is not moved in the caller's work area.

14 IGG0CLAZ: IGGPSHIN

The Locate routine puts the beginning address and the length of the catalog field into the CTGFL's field-data entry.

15 IGG0CLAZ: IGGPSHIN

The CTGFL's field-data entry contains the beginning address and length of the data in the caller's work area. When control is returned to the caller, the caller can use the field-data entry to locate a specific field's data in the work area.

16 IGG0CLAZ: IGGPSCNF

If the field name processed required the upgrade entry to be read in, the original entry is restored before the next CTGFL is processed.

17 IGG0CLAZ: IGGPEXT

If this function was requested by an internal catalog management function, final derived volume processing must be done. This processing consists of generating certain volume entry fields from the catalog information returned in the user's work area.

18 IGG0CLAZ: IGGPLOC

For additional information about topics related to LOCATE processing, see:

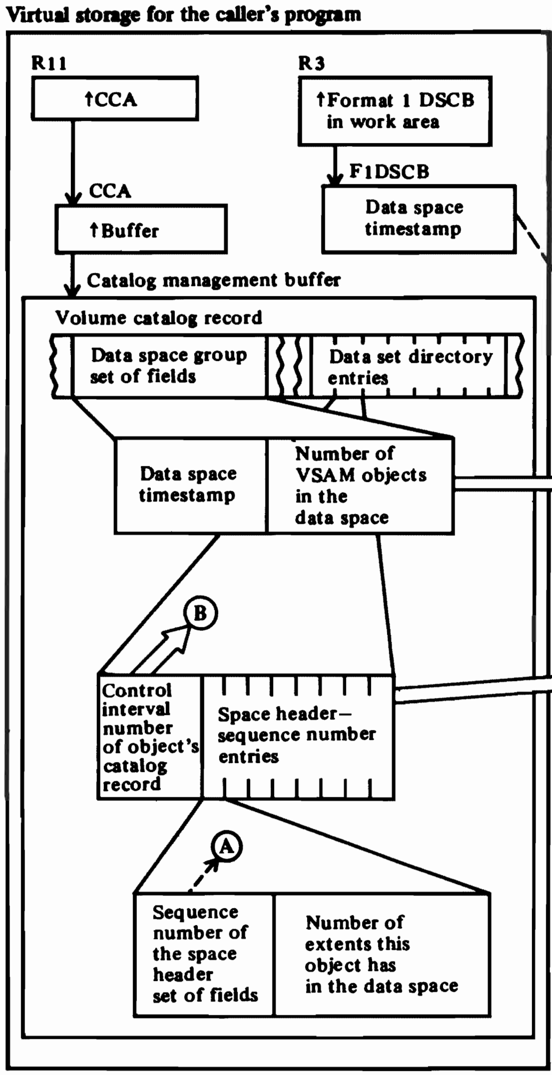
"Data Areas:"

Catalog parameter list (CTGPL) description and format
Field parameter list (CTGFL) description and format
Catalog communications area (CCA) description and format
Catalog record field names and descriptions

"Diagnostic Aids:"

Catalog management return codes

Diagram DF1. GENDSP: List the Contents of a Data Space



DB1 9

List all VSAM objects in a data space

1. Is data space unique?

No. Yes

2. Get the true-name record.
3. Put volume ACB address, CI number, and length of CI in user's area.
4. Return to caller

CCAPROB: Return code

5. Retrieve the data space group set of fields associated with the data space.

Do steps 6 through 8 to examine each Data Set Directory entry in the volume catalog record.

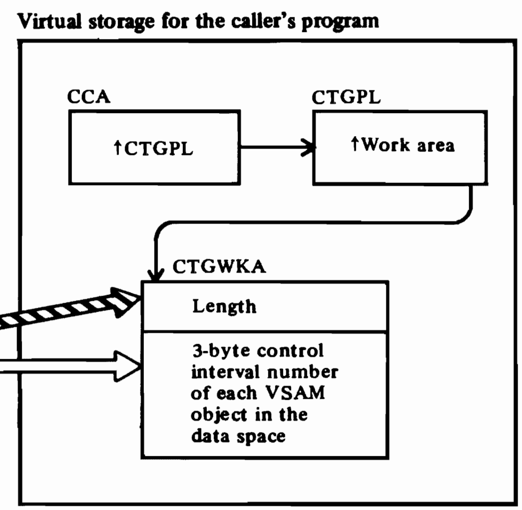
6. Retrieve the Data Set Directory entry.
7. Does the Data Set Directory entry indicate that its VSAM object is allocated an extent in the data space?

A

No Yes

B

8. Put the control interval number of the VSAM object's catalog record into the list.



9. Return to the caller when all Data Set Directory entries have been processed.

CCAPROB: Return code

Notes for Diagram DF1

The caller (an OS/VS Utilities program or OS/VS Open) specifies the GENDSP option of LOCATE to obtain the control interval number of the catalog record of each object (cluster, data set, index, and catalog, etc.) that is contained in a VSAM data space. The caller identifies the data space with its volume serial number and data space name (from the format-1 (identifier) DSCB).

1 IGG0CLBJ: IGGPGDSP

The user-provided work area is tested to ensure that the minimum size has been provided. The GENDSP routine tests the first seven characters of the data space name to determine whether the data space is unique. A data space name beginning with "Z999999..." is a nonunique data space.

2 IGG0CLBJ: IGGPGUDS

The true-name catalog record associates the data space name with the control interval number of the catalog record that describes the data space.

3 IGG0CLBJ: IGGPGUDS

The fixed length of the control interval number area and the control interval number are put into the user-provided work area.

5 IGG0CLBJ: IGGPGDSP

The CCA, a CPL, and two FPLs are set up to extract the data space group set of fields for the appropriate data space. The DSCB timestamp value is calculated from the data space name and used as the test value.

6 IGG0CLBJ: IGGPGDSP

Three FPLs are set up and the Data Set Directory entry is extracted.

7 IGG0CLBJ: IGGPGDSP

Scan the Data Set Directory set of fields to find a data space sequence number match.

8 IGG0CLBJ: IGGPGDSP

When a sequence number match is found, the volume ACB address, the length of the control interval number area, and the control interval number are put into the caller's work area.

For additional information about topics related to GENDSP processing, see:

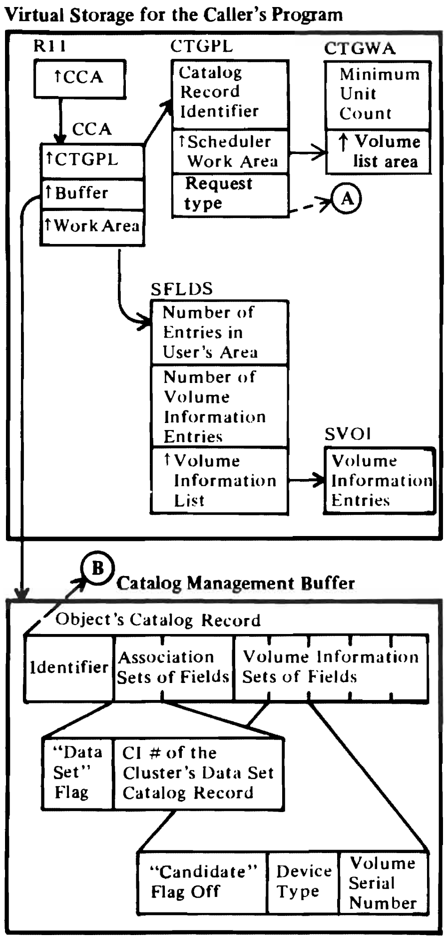
"Data Areas:"

Volume catalog record description and format
Fields derived from information in the volume catalog record
Volume catalog record sets of fields' descriptions and formats

"Diagnostic Aids:"

Catalog management return codes

Diagram DG1. SUPERLOCATE: List a Data Set's Volumes



DB1
9

List all volumes containing the caller-specified VSAM object.

1. Is the request LOCATE-by-generic-name or LOCATE-generation-data-group?

A

No Yes

2. LOCATE-by-generic-name: List all data set names (in a catalog) that match a user-supplied generic name (see Diagram DG5).

3. LOCATE-GDG: Build a generation data set's DSNAME that includes the data set's original DSNAME plus an absolute generation and version number (see Diagram DG6).

4. Are caller's work area and volume list area valid?

Yes No

5. Return to caller. CCAPROB Error code

5.1. Return catalog name, connector, and volser. D

6. Prevent the VSAM object from being deleted by another catalog management caller during the SUPERLOCATE process. C

7. Initialize CCA, CTGPL, and CTGFL to retrieve group associations of VSAM object.

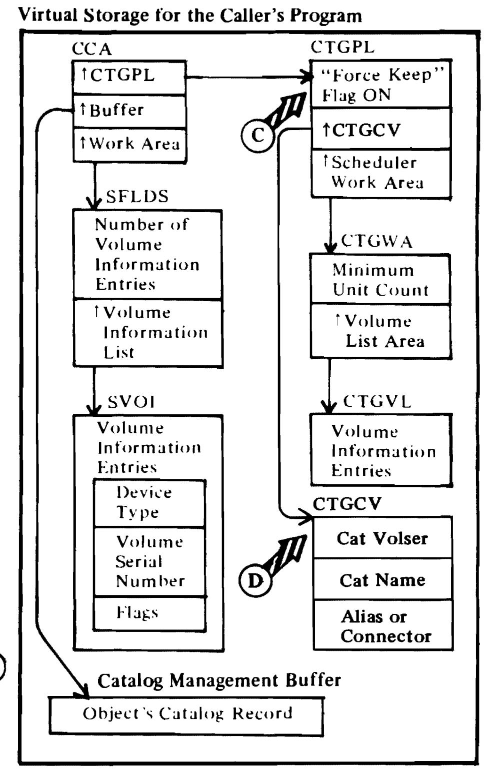
B

8. Is the VSAM object an index, data, cluster, nonVSAM, alternate index, path, or user catalog?

Yes No

9. Return to caller. CCAPROB Error code

DG2
10



Notes for Diagram DG1

The caller (the OS/VS Scheduler) specifies the SUPERLOCATE option of LOCATE to obtain a list of volume serial numbers and device types for a data set's volumes. The caller identifies the data set with its dsname value.

When the caller specifies LOCATE-by-generic-name, IGG0CLAM calls IGGPGLOC (IGG0CLAZ) to process the request. (See Diagram DG5 for LOCATE-by-generic-name processing details.) When the caller specifies a relative DSNAME (in the form "dsname(n)"), IGG0CLAM calls IGGPGDGL (IGG0CLAZ) to convert the relative DSNAME to an absolute DSNAME (in the form "dsname.GxxxxVyy") and locate the catalog record identified by the absolute DSNAME. (See Diagram DG6 for LOCATE-GDG-data-set processing details.)

1 IGG0CLAM: IGGPSLOC

4 IGG0CLAM: IGGPSLIN and **IGGPDBVC**

5.1 IGG0CLAM: IGGPSLIN

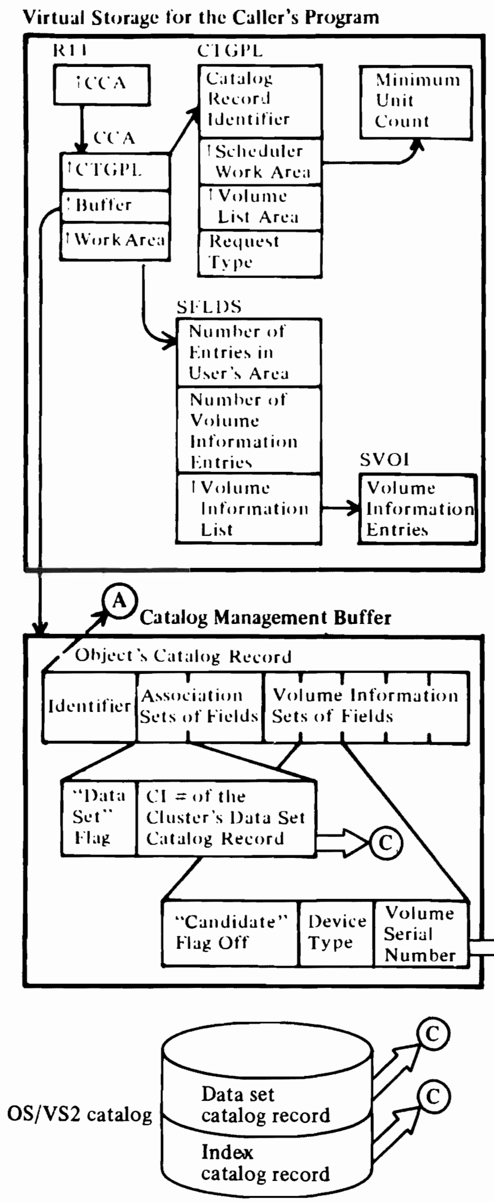
6 IGG0CLAM: IGGPSLIN

The CTGPL's "catalog record identifier" points to the object's dsname value in the caller's work area. The object might be a catalog, a VSAM data set, a nonVSAM data set, or a key-sequenced data set's index.

8 IGG0CLAM: IGGPSLOC

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DG2. SUPERLOCATE: List a Data Set's Volumes



DG1 8

(A) 10. Is the request in a recoverable catalog and the VSAM object is not a user catalog or nonVSAM entry?

No | Yes

11. Move catalog recovery area information to the volume list. (B)

(A) 12. Determine the type of object.

- Object is a cluster or alternate index entry.
- Object is a path entry. DG3 16
- Object is a nonVSAM, index, or user catalog entry. DG3 17
- Object is a data record. DG3 19

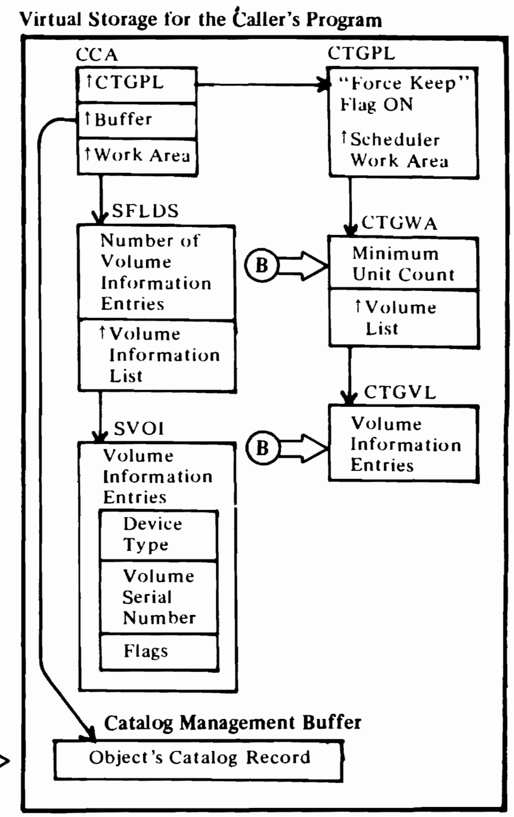
Cluster or Alternate Index Processing

(C) 13. Retrieve the base data and index (if any) records from each group occurrence of the VSAM object.

14. For each base record obtained in step 13 retrieve volume information for each volume that has space allocated to the object or is a candidate volume.

15. For each volume occurrence obtained in step 14, build a list of volume information entries and establish the minimum unit count. (B)

DG3 18



Notes for Diagram DG2

10 IGG0CLAA: IGGPSLEN and **IGGPSLIV**

12 IGG0CLAA: IGGPSLEN

13 IGG0CLAA: IGGPSLCG

If the VSAM object is a base cluster data record, call IGGPSLY (IGG0CLAA) to obtain upgrade associations, if any.

14 IGG0CLAM: IGGPSLEL

15 IGG0CLAA: IGGPSLIV

The volume list pointed to by CTGWAVL has the following format:

- The volume list contains no duplicate volume serial numbers.
- The volumes are divided by whether they are within the minimum unit count or outside it. Minimum unit count is the minimum number of direct-access devices required to mount the object's volumes. Volumes must be contiguous by device type. Device types within the minimum unit count are not ordered in any particular sequence nor are they related to the device types outside the count.
- Volumes within the minimum unit count will each be assigned an individual unit by the Scheduler. If volumes that do not have units already assigned exist outside the minimum unit count, the last unit assigned to a volume of the same device type within the minimum unit count will be made nonsharable. If this is not possible, an additional nonsharable unit will be assigned.

The volume list pointed to by CTGWAVL has the following content:

- All volumes in a given entry are placed into the volume list, regardless of whether they have allocated space.
- The volume information returned varies according to the entry type specified by the SUPERLOCATE request and whether the volume is within the minimum unit count or outside it, as follows:

  Entry Types C, D, G, I, and R

  Within the minimum unit count, the CRA volume for the particular entry is returned.

Data Entry (D):

Within the minimum unit count, every volume in the upgrade set is returned. Each volume in the data entry that has a unique device type within the data entry and is either the first with allocated space (prime or overflow) or, if no volumes have allocated space, is the first candidate volume returned.

Outside the minimum unit count, all others in the data entry are returned.

Index Entry (I):

Within the minimum unit count, each volume in the index entry that has a unique device type within the index entry and also is either the first with allocated space (prime or overflow) or, if there are no volumes with allocated space, is the first candidate volume returned. If sequence set is with data, the same volume may appear as both a prime and a candidate volume.

Outside the minimum unit count, all others in the index entry are returned.

NonVSAM Entry (A):

Within the minimum unit count, each volume in the nonVSAM entry that has a unique device type within the nonVSAM entry is returned. For single volume entries, the TTR DSCB pointer is also returned. Every nonVSAM volume occurrence is marked as prime. Outside the minimum unit count, all others in the nonVSAM entry are returned.

User Catalog Entry (U):

Within the minimum unit count, each volume in the user catalog entry that has a unique device type within the entry is returned. Every user catalog volume occurrence is marked as prime.

Outside the minimum unit count, all others in the user catalog entry are returned.

Base Cluster Entry (C):

Within the minimum unit count, every volume that does not have sequence set with data is returned. Otherwise, same as data entry.

Outside the minimum unit count, all others in the data entry are returned.

Alternate Index Entry (G):

Same as the base cluster entry, except that there is never an upgrade set.

Alias Path Entry (R):

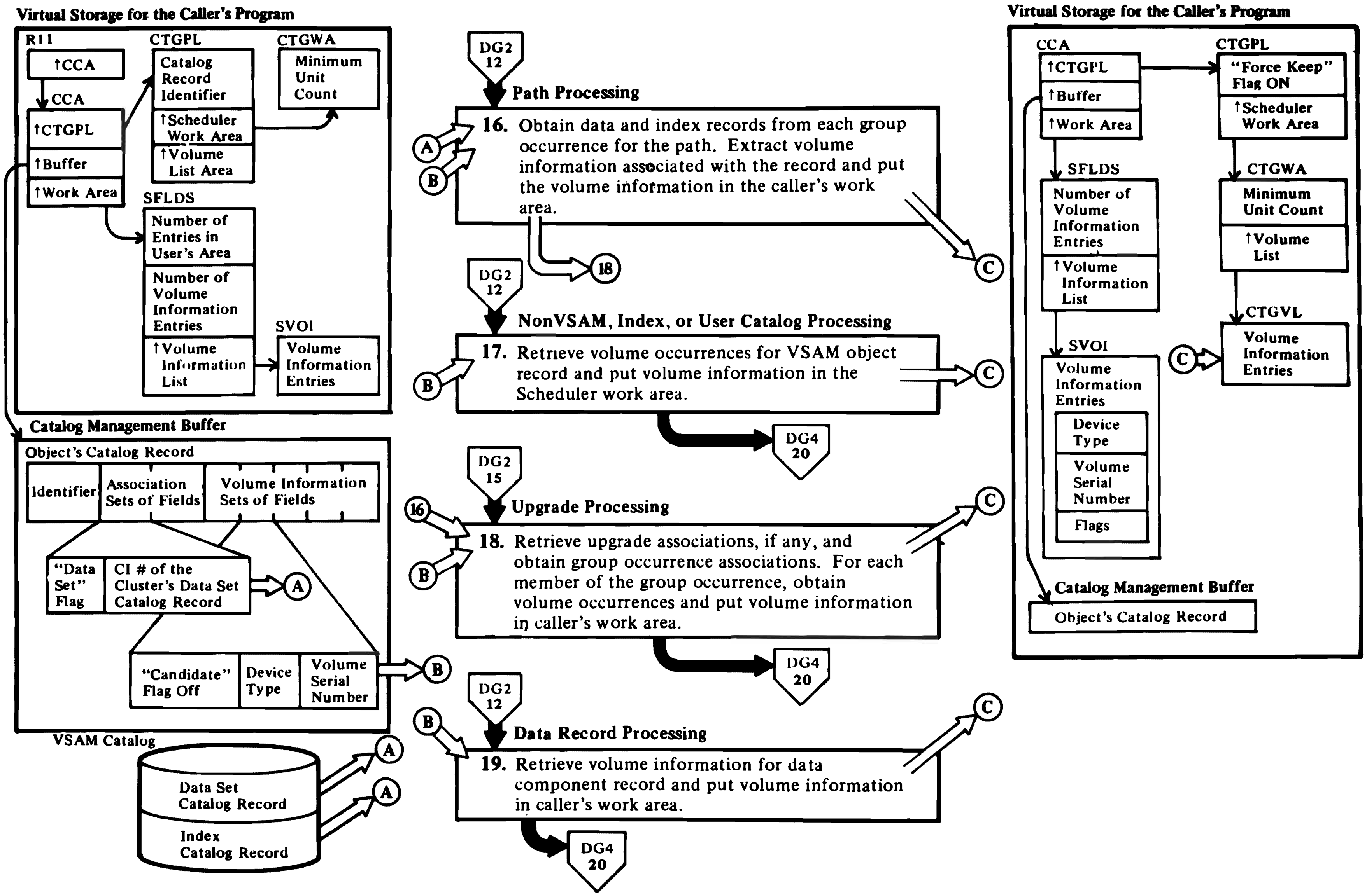
Same as the base cluster entry, except that upgrade set inclusion depends on the UPDATE/NOUPDATE flag in the path entry.

Normal Path Entry (R):

Within the minimum unit count, every volume of the alternate index under this path is returned. Otherwise, same as the base cluster entry, except that upgrade set inclusion depends on the UPDATE/NOUPDATE flag in the path entry.

Outside the minimum unit count, all others in the data entry are returned.

Diagram DG3. SUPERLOCATE: List a Data Set's Volumes



Notes for Diagram DG3

16 IGG0CLAA: IGGPSLR, IGGPSLIV, and **IGGPSLY**

For each record obtained, IGGPSLEL extracts the volume information associated with the record. Then IGGPSLIV inserts the information into the caller's work area. When the base cluster data record has been retrieved, IGGPSLY obtains any upgrade associations related to the record.

17 IGG0CLAA: IGGPSLIV

IGG0CLAM: IGGPSLEL

IGGPSLIV inserts volume information into the caller's work area.

18 IGG0CLAA: IGGPSLY and **IGGPSLIV**

IGG0CLAM: IGGPSLEL

IGGPSLEL obtains the volume occurrences; IGGPSLIV inserts the volume information into the caller's work area.

19 IGG0CLAA: IGGPSLEN

If the VSAM object is a base cluster data record, call IGGPSLY (IGG0CLAA) to obtain upgrade associations, if any.

Diagram DG4. SUPERLOCATE: List a Data Set's Volumes

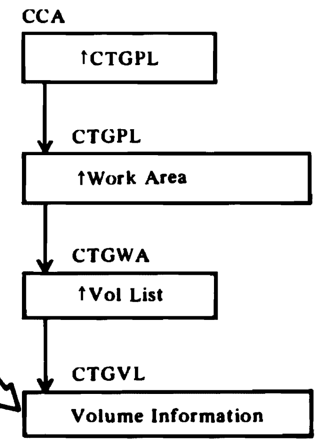
DG3 17 | DG3 18 | DG3 19

20. Move volume information to beginning of caller's work area.

21. Return to caller.

CCAPROB

Return Code



Notes for Diagram DG4

20 IGG0CLAA: IGGPSLEN and **IGGPSLIV**

IGGPSLIV builds the volume list from the end of the work area to the beginning, thus allowing the sorting of entries by device type both within and outside the minimum unit count.

21 IGG0CLAM: IGGPSLOC

If an error is detected, the procedure detecting the error returns control immediately to the calling procedure. IGGPSLOC returns to the caller of SUPERLOCATE.

If no error is detected and the request was for a single, nonVSAM entry, the data set's DSCB TTR is returned to the caller. IGGPSLEL (IGG0CLAM) sets the indicator for returning the TTR and saves the DSCB TTR in the SUPERLOCATE work area.

For additional information about topics related to SUPERLOCATE processing, see:

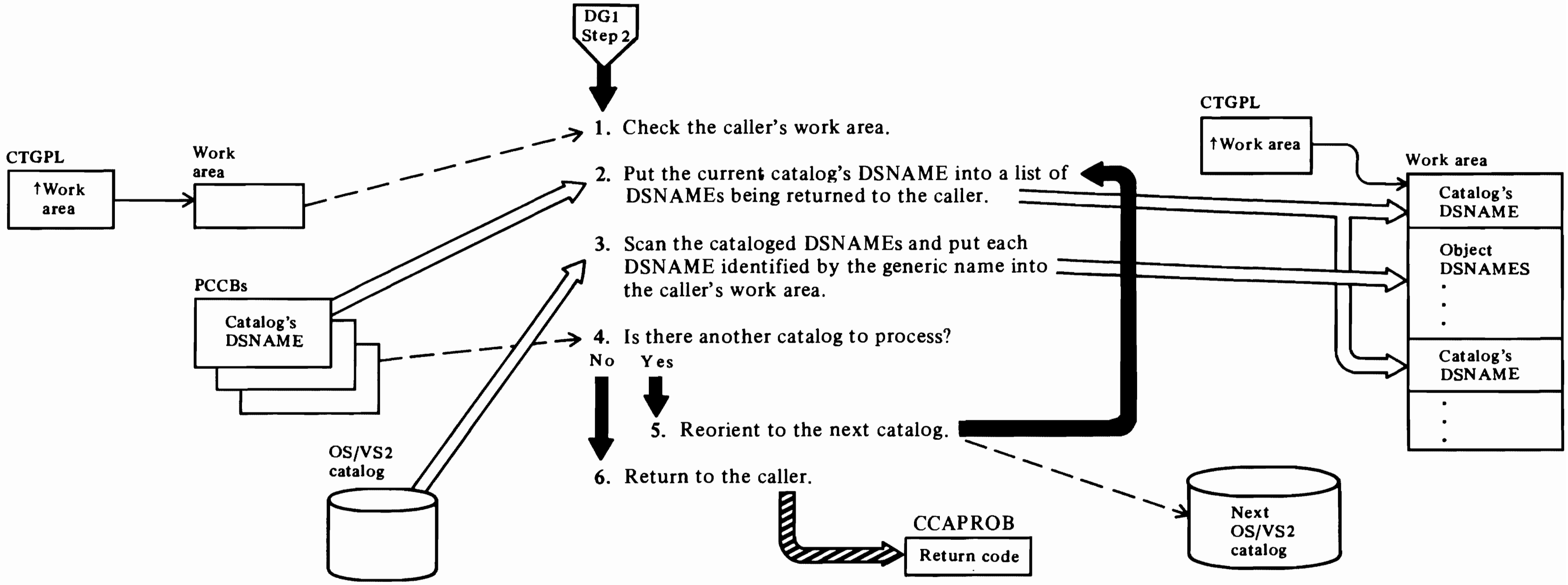
"Data Areas:"

Catalog record descriptions and formats

"Diagnostic Aids:"

Catalog management return codes

Diagram DG5: LOCATE Based on a Generic Name



Notes for Diagram DG5

When the caller specifies LOCATE-by-generic-name, IGGPGLOC is called to process the request. The caller's CTGPL points to a work area. When IGGPGLOC returns to the caller, the work area contains a list of cataloged data set DSNAMEs identified by the generic name.

1 **IGG0CLA2: IGGPGLOC** (calls **IGGPDBVC (IGG0CLAM)**)

2 **IGG0CLA2: IGGPGLCN**

The DSNAME of the catalog being searched is put into the caller's work area. Each cataloged-object DSNAME identified by the caller-supplied generic name is then put into the caller's work area.

3 **IGG0CLA2: IGGPGLOC**

4 **IGG0CLA2: IGGPGLOC**

The caller specifies the catalogs to be searched for DSNAMEs identified by the generic name.

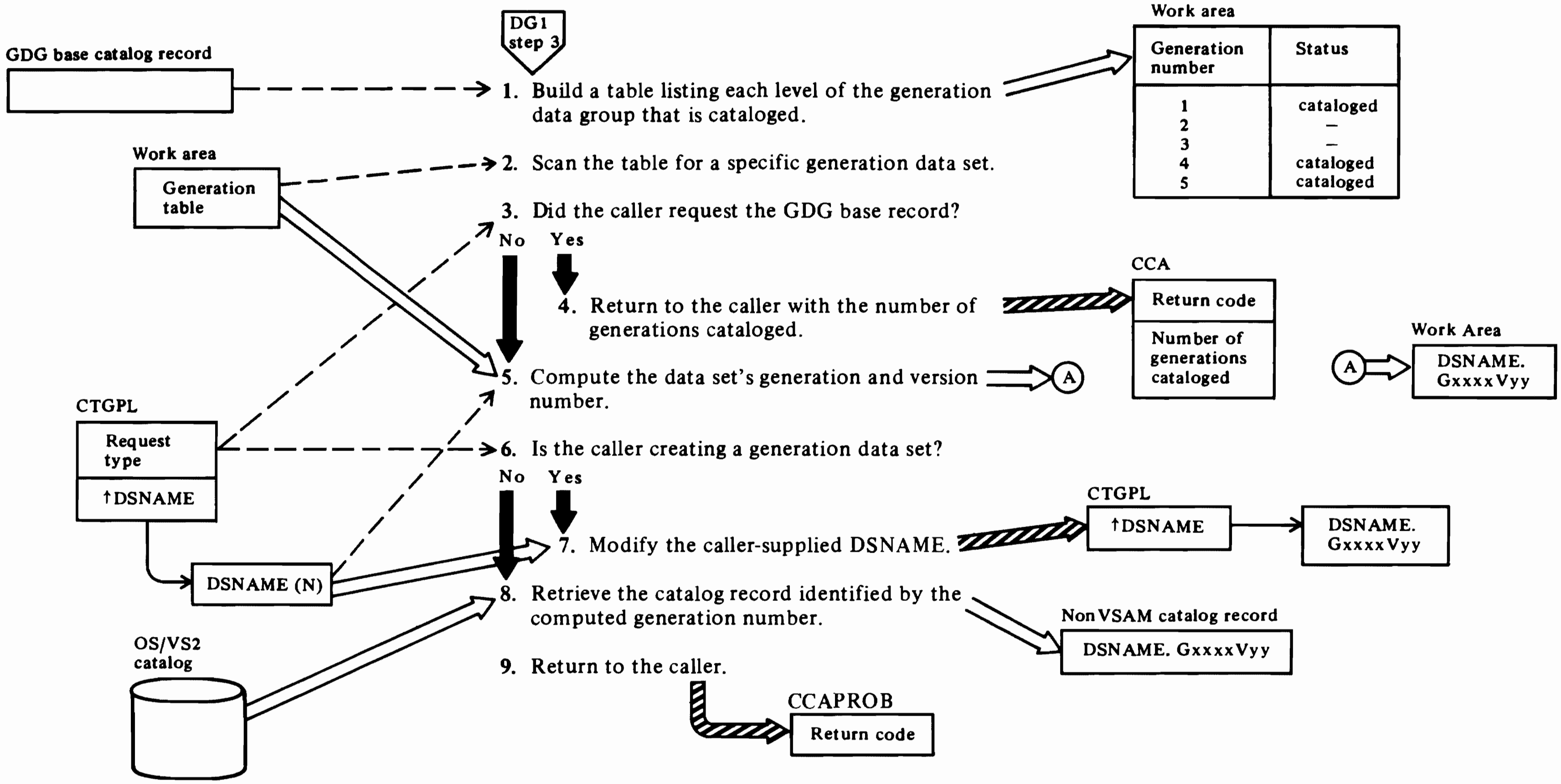
5 **IGG0CLA2: IGGPGLOC** (calls **IGGPSC2 (IGG0CLA3)**)

IGGPSC2 searches the PCCB chain to find the next caller-specified catalog and connect the caller's request to it. If the specified catalog isn't open, IGGPSC2 opens it.

6 **IGG0CLA2: IGGPGLOC**

If the caller's work area isn't large enough to contain all of the DSNAMEs found, IGGPGLOC sets the appropriate return code and puts the required length in the CCA.

Diagram DG6: Locate a Generation Data Set's Catalog Record



1. Build a table listing each level of the generation data group that is cataloged.
2. Scan the table for a specific generation data set.
3. Did the caller request the GDG base record?
4. Return to the caller with the number of generations cataloged.
5. Compute the data set's generation and version number.
6. Is the caller creating a generation data set?
7. Modify the caller-supplied DSNAME.
8. Retrieve the catalog record identified by the computed generation number.
9. Return to the caller.

Notes for Diagram DG6

When the caller specifies a relative DSNAME (in the form "dsname(n)") that identifies a generation data set (nonVSAM), IGGPGDGL is called to locate the record identified by the DSNAME. The caller specifies "dsname(0)' to locate the latest generation data set, "dsname(-1)' to locate the next latest generation data set, etc. The caller can also specify a starting point, or "zero level', generation. The zero-level generation thus defined becomes the latest generation data set.

If the generation data set is being created, IGGPGDGL is called to convert the relative DSNAME to an absolute DSNAME in the form "dsname.GxxxxVyy". The OS/VS2 Scheduler encounters the relative DSNAME in a JCL statement (the relative DSNAME is usually "dsname(+1)".) The OS/VS2 Scheduler issues a SUPERLOCATE command to catalog management, pointing to the relative DSNAME. Catalog management returns to the Scheduler with the data set's absolute DSNAME in the form "dsname.GxxxxVyy". Subsequently, the OS/VS2 Scheduler issues a DEFINE NONVSAM command to create a nonVSAM (generation) data set with the DSNAME "dsname.GxxxxVyy".

1 IGG0CLA2: IGGPGTAB

The GDG base catalog record contains a variable-length header field. This field consists of 1- or 2-byte entries that identify each generation data set in the group and specify whether the data set is cataloged.

2 IGG0CLA2: IGGPGRGN

If the caller specified a zero-level generation, IGGPGRGN updates the generation level table to reflect the new zero-level generation.

4 IGG0CLA2: IGGPGFND

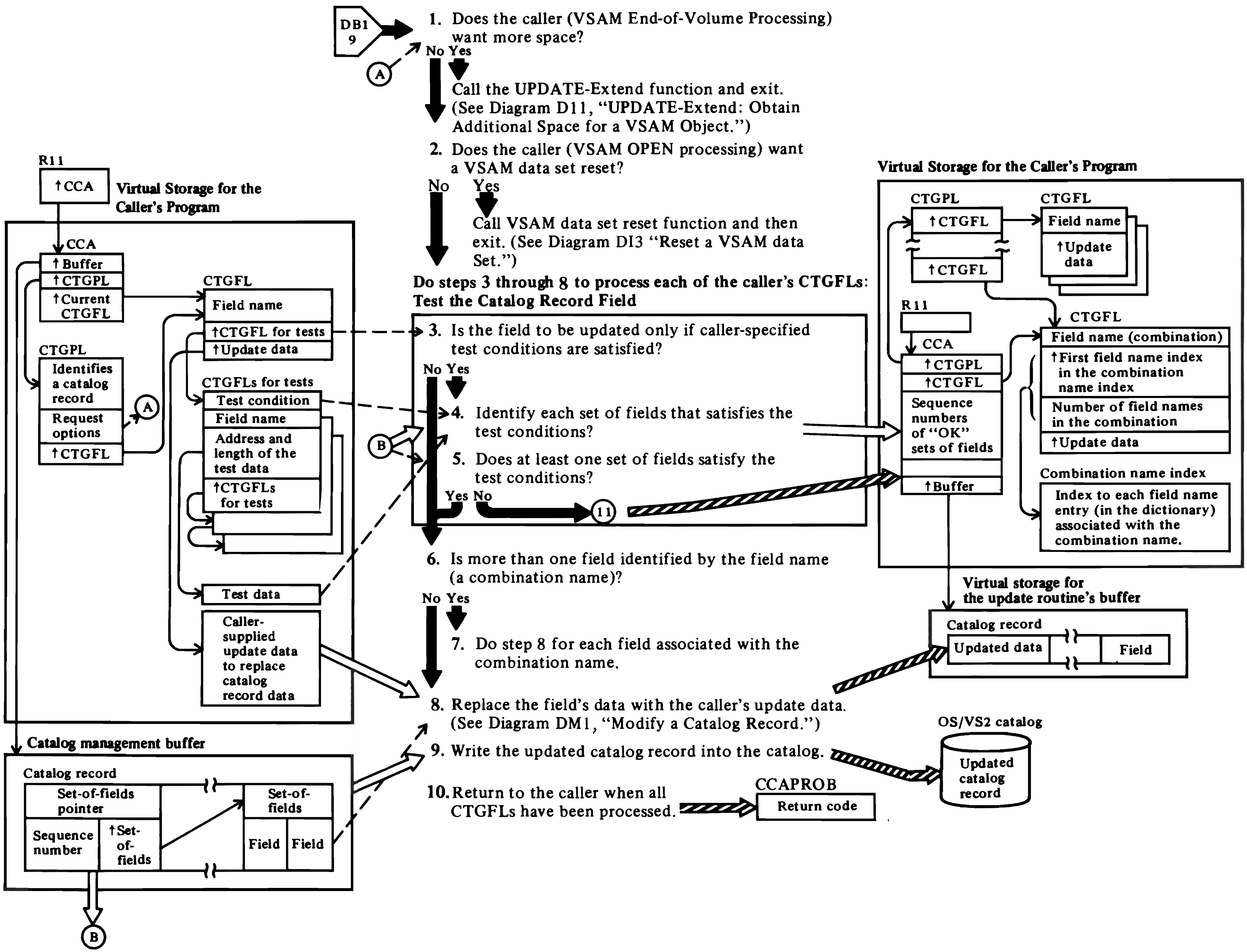
If the caller didn't specify a generation data set (the caller specified an unqualified DSNAME), IGGPGFND determines the number of generations that are cataloged and returns this information to the caller.

5 IGG0CLA2: IGGPGDGL

The caller-supplied relative DSNAME is in the form 'dsname(n)' where 'n' identifies the generation data set. The generation data set's cataloged DSNAME is in the form "dsname.GxxxxVyy' where "Gxxxx' is the generation number and "Vyy' is the version number.

8 IGG0CLA2: IGGPGGE

Diagram DH1. UPDATE: Modify Catalog Information



1. Does the caller (VSAM End-of-Volume Processing) want more space?
   No Yes
   Call the UPDATE-Extend function and exit. (See Diagram DI1, "UPDATE-Extend: Obtain Additional Space for a VSAM Object.")
2. Does the caller (VSAM OPEN processing) want a VSAM data set reset?
   No Yes
   Call VSAM data set reset function and then exit. (See Diagram DI3 "Reset a VSAM data Set.")

Do steps 3 through 8 to process each of the caller's CTGFLs: Test the Catalog Record Field

3. Is the field to be updated only if caller-specified test conditions are satisfied?
   No Yes
4. Identify each set of fields that satisfies the test conditions?
5. Does at least one set of fields satisfy the test conditions?
   Yes No
6. Is more than one field identified by the field name (a combination name)?
   No Yes
7. Do step 8 for each field associated with the combination name.
8. Replace the field's data with the caller's update data. (See Diagram DM1, "Modify a Catalog Record.")
9. Write the updated catalog record into the catalog.
10. Return to the caller when all CTGFLs have been processed.

Notes for Diagram DH1

IDA0200T

The VSAM Close routine uses VSAM catalog management to modify the data set and index statistics maintained in the catalog record's copy of the AMDSB.

See VSAM Close Processing, in *OS/VS2 VSAM Logic*, for details about disconnecting a user from a VSAM cluster (Close processing).

IDA0557A

The VSAM EOV routine uses VSAM catalog management to obtain more space for a data set.

See VSAM End-of-Volume Processing, in *OS/VS2 VSAM Logic*, for details about end of volume processing.

IDA0192A

The VSAM Open routine uses VSAM catalog management to reset a VSAM data set. See "VSAM Open Processing" in *OS/VS2 VSAM Logic* for details about Open Processing.

IGG0CLAB: IGGPACDV (calls **IGGPUPD (IGG0CLAV)**)

When the caller issues the CATLG macro instruction, register 1 points to the caller's catalog parameter list (CTGPL). The CTGPL request options are decoded and the base catalog record is retrieved for the request. See Diagram DB1, VSAM Catalog Management Overview, for a description of initial catalog management processing and request decoding.

1 **IGG0CLAV: IGGPUPD** (calls **IGGPUPDE (IGG0CLBB)**)

If more space is required for the data set, the UPDATE—Extend routine processes the caller's update request, and returns to IGGPACDV in IGG0CLAB.

2 **IGG0CLAV: IGGPUPD** (calls **IGGPRUS (IGG0CLB7)**)

If a VSAM data set must be reset, the UPDATE-Reuse routine processes the caller's update request and returns to IGGPACDV (IGG0CLAB).

IGG0CLAV: IGGPSFPL

Steps **3** through **8** are performed to update each of the catalog record fields identified by the caller's CTGFLs.

3 **IGG0CLAV: IGGPSFPL**

The caller's CTGFL list contains the address of each CTGFL needed to satisfy the caller's updating requirements. Each field parameter list (CTGFL) describes one of the catalog record fields to be updated. Each CTGFL is completely processed before the next one is started.

IGG0CLAV: IGGPSFPL (calls **IGGPTSTS (IGG0CLBA)**)

The caller might want to update a field only if another field's value, when compared to the caller's test value, satisfies the caller's test conditions. If so, the caller builds a CTGFL that contains the name of the catalog field to be tested, the test conditions (equal, high, low, etc.), and the address and length of the caller's test value. If a CTGFL contains the address of another CTGFL, the second CTGFL describes a catalog record field that is to be compared to the caller's data. If the comparison satisfies the test conditions, the catalog record field specified by the first CTGFL is updated with the caller's data.

4 **IGG0CLBA: IGGPTSTS** and **IGGPTCMP**

If the caller wants to update a catalog record's header field, the field's data is updated with the caller's data if all tests are satisfied.

If the caller wants to update a field from one of the sets of fields that follow the header field, the field's data is updated with the caller's data for each set of fields' field that satisfies all tests. The set of fields that contains the field to be updated can also be identified by its sequence number.

The sequence number of each set of fields that satisfies the tests is put in the CCA. When all sets of fields have been tested, the sequence numbers are used to identify each set of fields that contains caller-requested data.

6 **IGG0CLAX: IGGPALT2**

A combination name refers to a set of related catalog field names, and is used by the caller instead of a separate CTGFL for each field name.

7 **IGG0CLAX: IGGPALT2**

The CCA's combination name index has an entry for each field name in the combination. The Update routine processes each field name entry in the combination name index sequentially, starting with the index of the first field name entry for the combination, and ending when the number of entries processed equals the number of field names associated with the combination name.

The combination name's CTGFL contains the beginning address and the total length of the group of update data fields in the caller's work area.

The test sequence (if any) associated with a combination-name CTGFL is done only once, not once for each field name in the combination.

9 **IGG0CLAV: IGGPSFPL** (calls **IGGPPREC (IGG0CLAW)**)

When the catalog record is updated (in a buffer in the Update routine's virtual storage) the Update routine sets the "must write" flag on to indicate that the buffer must be written from virtual storage into the catalog before the buffer can be made available to contain another catalog record. When the caller's update request is finished, or when the Update routine needs the buffer to process another catalog record associated with the request, the Update routine calls IGGPPUPC or IGGPPAD (IGG0CLAG) to write the catalog record from the buffer into the VSAM catalog (on a direct-access storage device).

For additional information about topics related to UPDATE processing, see:

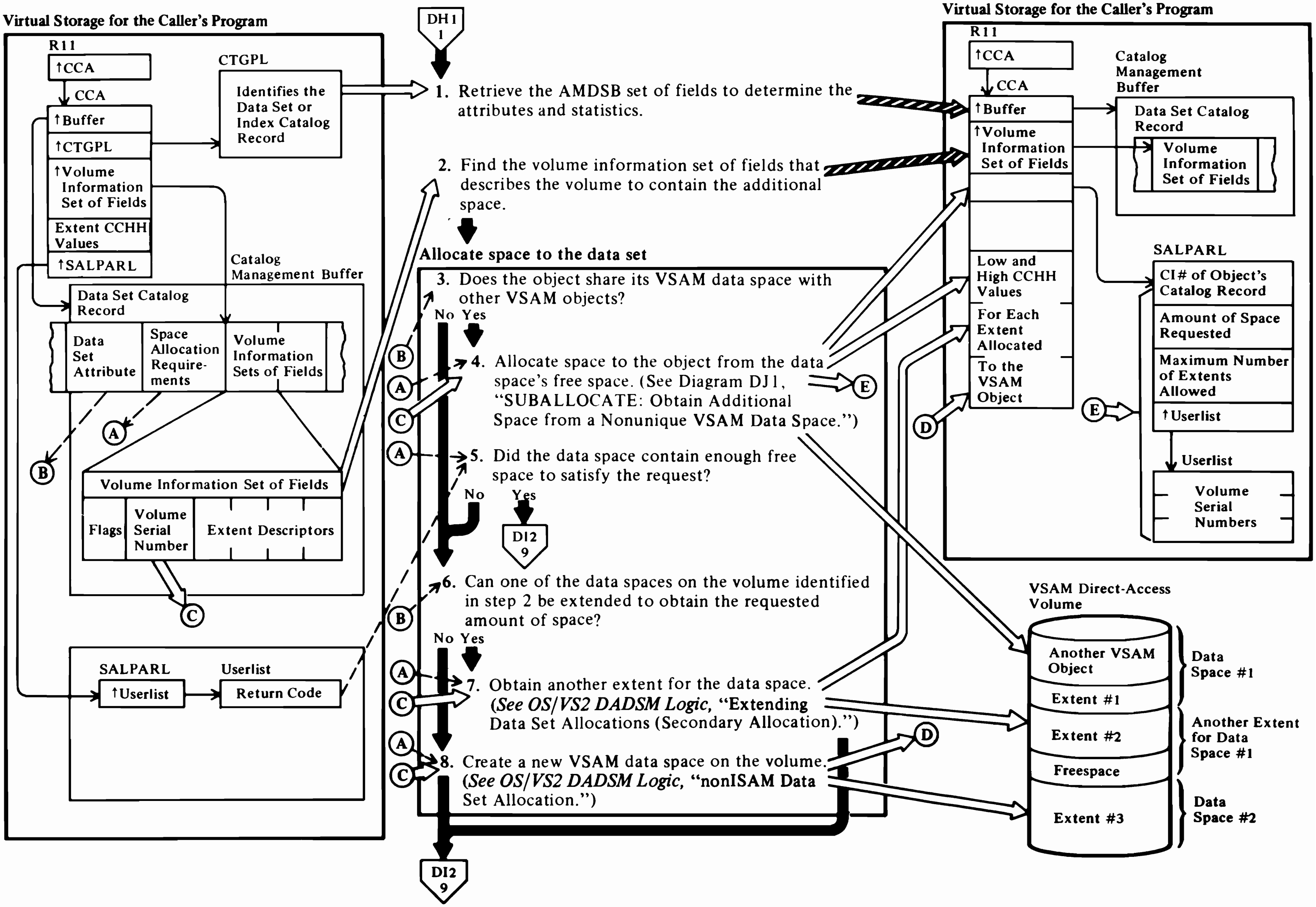
"Data Areas:"

Catalog record descriptions and formats
Catalog parameter list (CTGPL) description and format
Field parameter list (CTGFL) description and format
Catalog communications area (CCA) description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram DI1. UPDATE-Extend: Obtain Additional Space for a VSAM Object



Notes for Diagram DI1

The UPDATE-Extend routine is called whenever a VSAM object (cluster, data set, index, or catalog) needs more space to store its records.

The VSAM end of volume routine calls the catalog-management Update routine, and an amount of space, based on the object's direct-access space allocation requirements, is allocated from one of the following sources:

- A shared VSAM data space that has enough free space to satisfy the allocation requirements
- A shared VSAM data space, extended to satisfy the allocation requirements from the free space on the object's currently mounted volume
- A new VSAM data space, created to satisfy the allocation requirements, on the object's currently mounted volume.

See VSAM End-of-Volume Processing, in *OS/VS2 VSAM Logic*, for EOV processing details.

1 IGG0CLBB: IGGPUPDE (calls **IGGPINIT (IGG0CLBC)**)

2 IGG0CLBB: IGGPUPDE (calls **IGGPSVOL (IGG0CLBC)**)

The volume information set of fields is identified by volume serial and key ranges, if this is a key-range data set.

3 IGG0CLBB: IGGPUALL

A shared (nonunique) VSAM data space contains all or parts of two or more VSAM objects. A unique VSAM data space contains all or part of only one VSAM object, and is not allowed to contain records of another object.

4 IGG0CLBB: IGGPCSAL (calls **IGGPSALL (IGG0CLAR)**)

If the object shares its data space with other VSAM data sets or indexes, there might be enough free space in one of the data spaces on the volume to satisfy the object's direct-access space allocation requirements.

5 IGG0CLBB: IGGPUPDE

If there is not enough free space, another extent is obtained for one of the volume's data spaces, or a new data space is created.

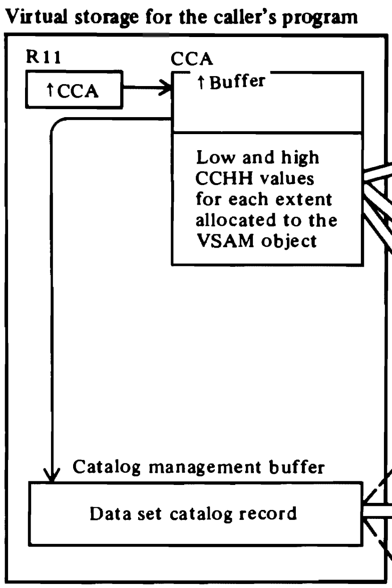
6 IGG0CLBB: IGGPUPDE

If any data space on the volume has less than 12 extents, the data space can be extended.

8 IGG0CLBB: IGGPUPDE

If a new extent was obtained for one of the volume's data spaces or a new data space was created, this space is suballocated to the object (nonunique) or given directly to the object (unique). For recoverable catalogs, the data set directory of the object is updated with a new sum total track value. For nonrecoverable catalogs, the format-4 timestamp field is updated on the physical volume.

Diagram DI2. UPDATE – Extend: Obtain Additional Space for a VSAM Object



DI1 7,8 — DI1 5

9. Compute the extent values, determine the new high RBA, and calculate the extent range for each extent in the newly allocated space.

10. Update the statistics in the AMDSB set of fields.

11. Does the volume information set of fields located by step 2 describe a key range volume to be added?

No — Yes

12. Add extend descriptors to a new volume information set of fields. (See Diagram DM1, "Modify the Catalog Record"– Add a New Set of Fields to the Record.)

13. Update the volume information set of fields to show the object's newly obtained space. (See Diagram DM2, "Modify the Catalog Record"– Change a Field Within a Set of Fields.)

Allow for sequence set stored with the data

14. If the object is key-sequenced, is the sequence set stored with the data in the data set?

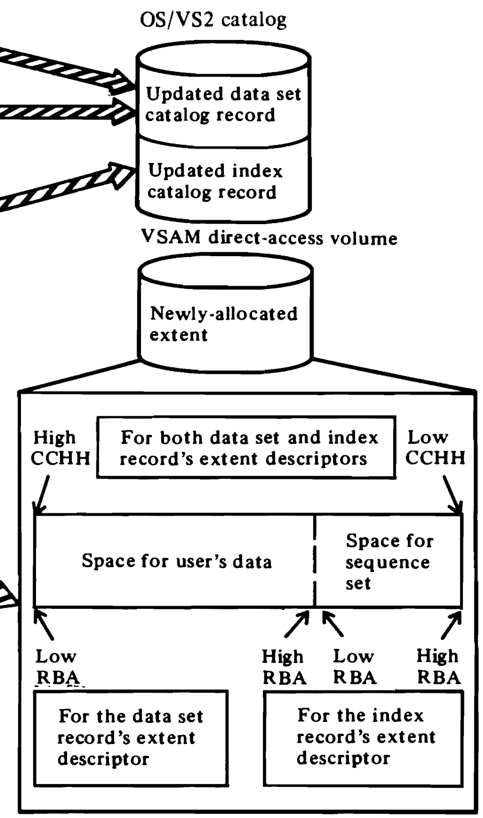
No — Yes

15. Retrieve the index catalog record.

16. Allocate part of the data set's new extent to the sequence set.

17. If SMF is required, write SMF record type 63– VSAM data set altered–to reflect the new extents and record type 69–VSAM data space allocated or extended. SMF records are not written for CRA data sets.

18. Return to the caller. → CCAPROB: Return code



Notes for Diagram D12

11 IGG0CLBB: IGGPMVOL

If a key range data set obtains space from a candidate volume, the second, third, fourth, etc., key range that obtains space from that candidate volume for the first time will require a new volume information set of fields. Note that each key range on each different volume is described by an individual volume information set of fields.

12 IGG0CLBB: IGGPMVOL

The object's catalog record contains a volume information set of fields to describe the object's space on each volume that contains a part of the object. If the object's newly obtained extent is in a key range on a new volume, the UPDATE-Extend routine may build (see step **11**) a volume information set of fields to describe the new volume and extent. Otherwise, an existing volume information set of fields is updated with the high-allocated RBA and extent information in the form:

SS CCHH CCHH DDDD DDDD

where:

SS identifies the VSAM data space.

CCHH are the low and high cylinder and track addresses.

DDDD are the low and high RBAs.

16 IGG0CLBB: IGGPSSWD

The low and high CCHH addresses (in the index catalog-record's volume information set of fields) are those of the extent obtained for the data set. The low and high RBA values are for the sequence set.

For additional information about topics related to UPDATE-Extend processing, see:

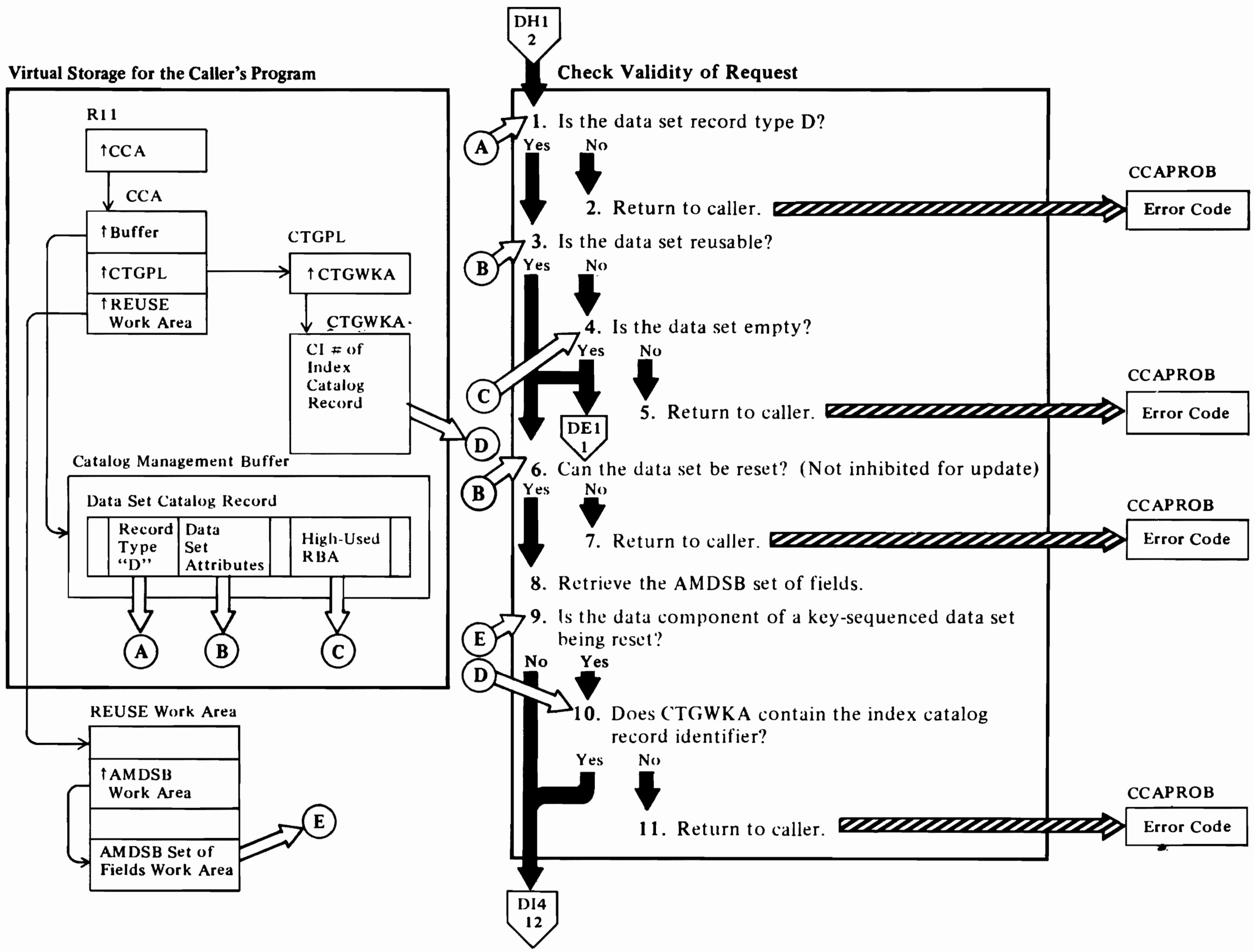
"Data Areas:"

Catalog record descriptions and formats
Volume information set of fields description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram DI3. REUSE: Reset a VSAM Data Set



Notes for Diagram DI3

Reusable is an attribute that may be assigned to a VSAM data set or an alternate index. This attribute allows the data set to have its high-used RBA set to zero at open time, if the user specifies the RESET option. The indicator for REUSE is retained in the attributes field of data and index records of a key-sequenced data set and an alternate index and in the data record of an entry-sequenced and relative record data set. Reusable data sets may be multi-volumed and they must be suballocated only. That is, reusable data sets may not be unique. Also, reusable data sets cannot have key ranges and they are restricted to a maximum of 16 physical extents per volume. If a base cluster is defined as reusable, it may not have alternate indexes associated with it; however, it is permissible to define reusable alternate indexes that are related to a nonreusable base cluster.

1 IGG0CLAV: IGGPUPD (calls **IGGPRUS (IGG0CLB7)**)

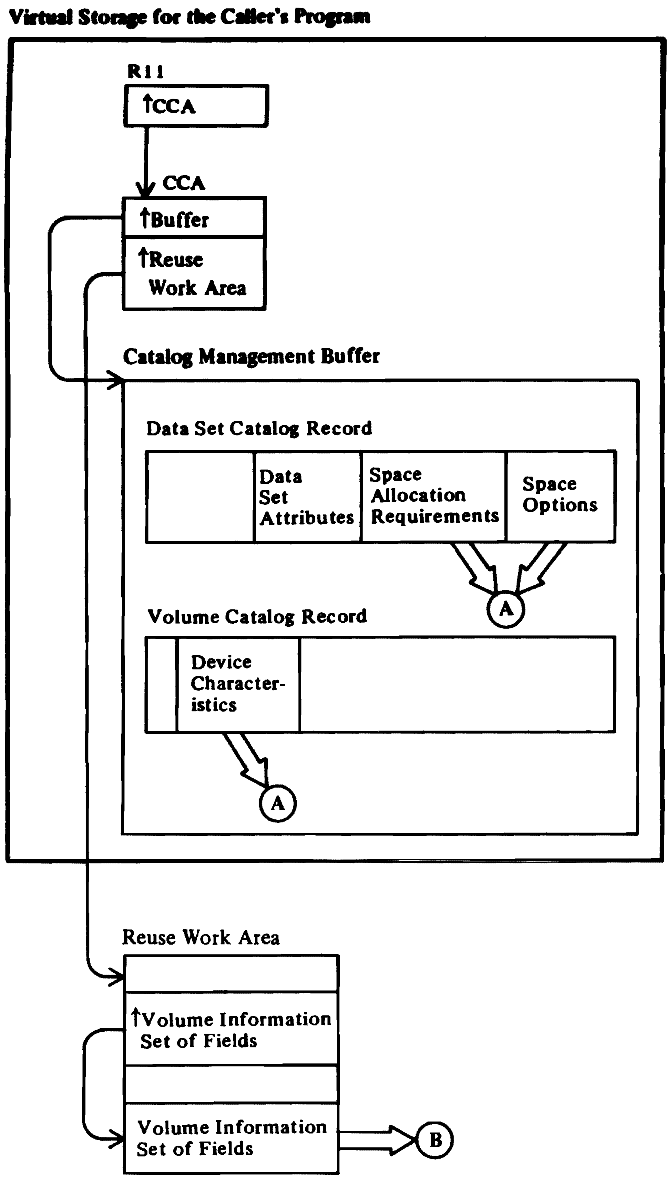
Initially, the data set catalog record is available in a catalog management buffer.

4 If the data set is not reusable but is empty, the caller's LOCATE request is processed. No error codes are returned.

8 IGG0CLB7: IGGPRUS (calls **IGGPEXT (IGG0CLAZ)**)

The reuse work area is set up by IGGPRUS and appropriate pointers are initialized. The reuse work area includes CTGPL and CTGFLs required by Modify and Extract logic.

Diagram D14. REUSE: Reset a VSAM Data Set



DI3 9,10

Reset Volume Information Set of Fields

12. Retrieve a volume information set of fields into the reuse work area.
13. Read corresponding volume record into the catalog management work area.
14. Is it the first noncandidate volume to be processed?

No Yes

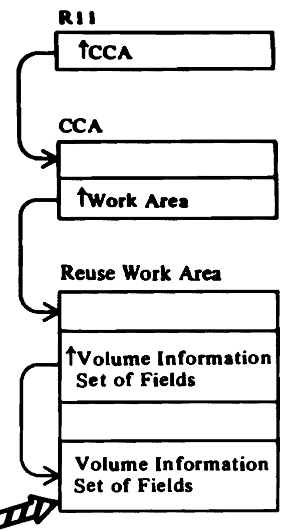
(A) 15. Calculate the high-used RBA for primary allocation of the data set.

(B) 16. Update volume information set of fields, retaining only those extents within the primary allocation. If no extents remain, the volume becomes a candidate volume.
17. Move updated volume information set of fields to the catalog (see Diagram DM1. "Modify a Catalog Record Field's Value").
18. Were any secondary extents present in the original volume information set of fields?

No Yes

DI5 25

DI5 19



Notes for Diagram DI4

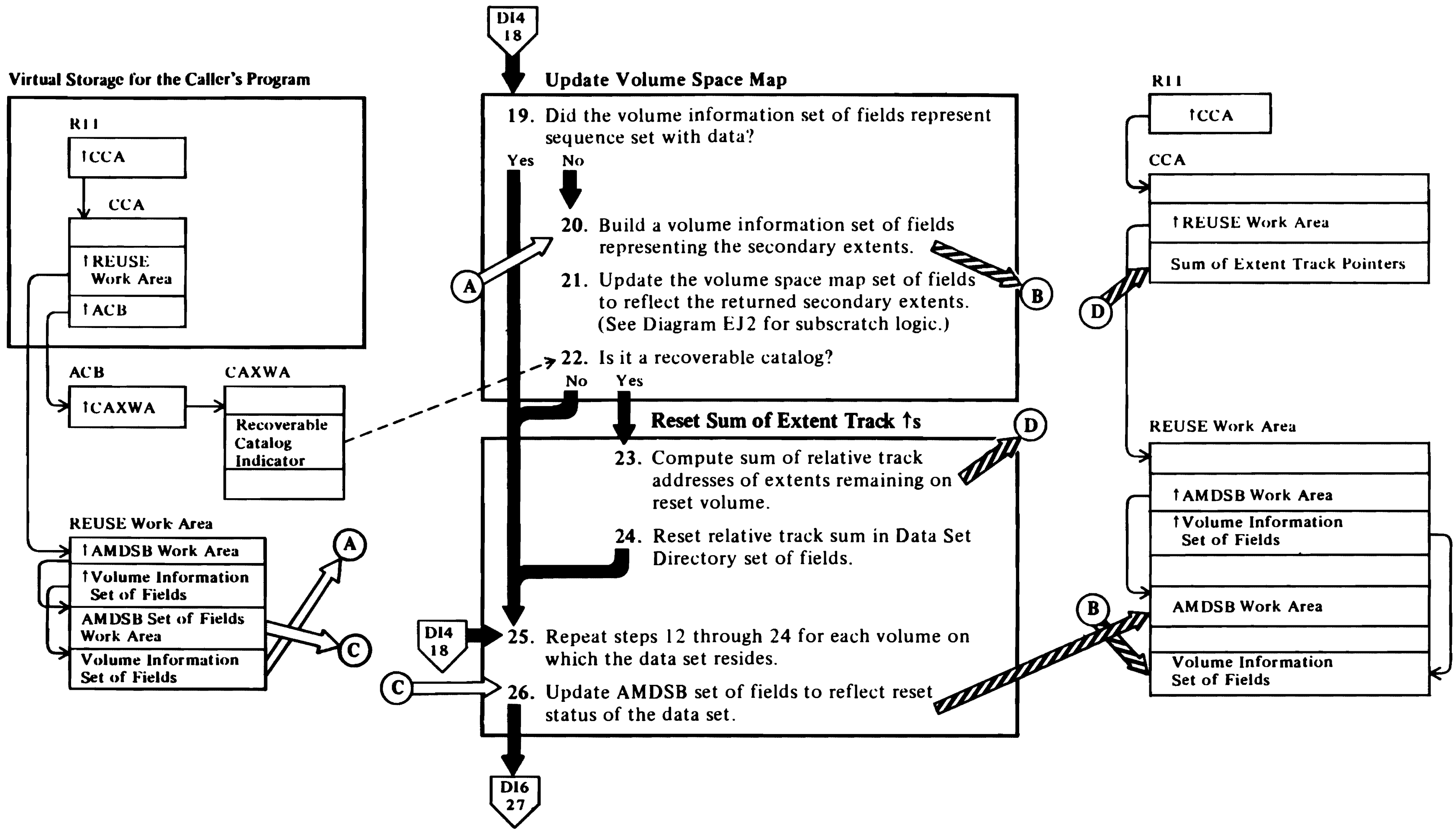
12 IGG0CLB7: IGGPRUS (calls **IGGPEXT (IGG0CLAZ)**)

The reuse work area provides space for a volume information set of fields that contains a maximum of 16 extent descriptors.

13 IGG0CLB7: IGGPRUS (calls **IGGPGET (IGG0CLBI)**)

17 IGG0CLB7: IGGPRUS (calls **IGGPMOD (IGG0CLAV)**)

Diagram DI5. REUSE: Reset a VSAM Data Set



Notes for Diagram DI5

19 The flag field in the volume information set of fields indicates whether sequence set is with data. If so, no resetting of the space map is necessary, since the space map will have been set already from the corresponding data set volume information set of fields. This is also true for the relative track sum in the case of a recoverable catalog.

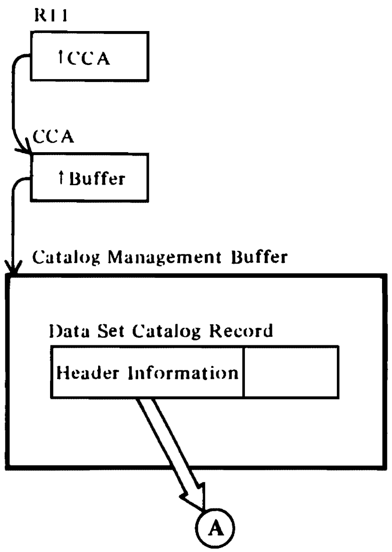
21 **IGG0CLB7: IGGPRUS** (calls **IGGPSSCR (IGG0CLBF)**)

23 **IGG0CLB7: IGGPRUS** (calls **IGGPTNXO (IGG0CLBI)**)

If the reset volume is a candidate volume, the computed sum in the CCA will be zero.

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DI6. REUSE: Reset a VSAM Data Set



DI5
26

A

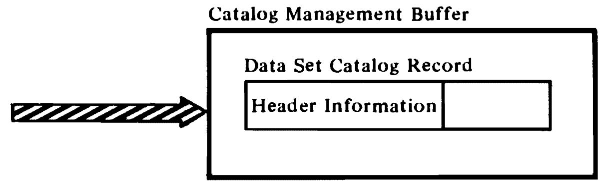
27. Update header information in data set catalog record to reflect reset status.
28. Move updated AMDSB sets of fields and updated header information to catalog.
29. Is a key-sequenced data set being processed?

No Yes

30. Read index catalog record into catalog management buffer, replacing the data set catalog record.
31. Repeat steps 8 through 28 for the index component of the data set.
32. Restore the data set catalog record into the catalog management buffer.

33. Return to caller.

CCAPROB
Return Code



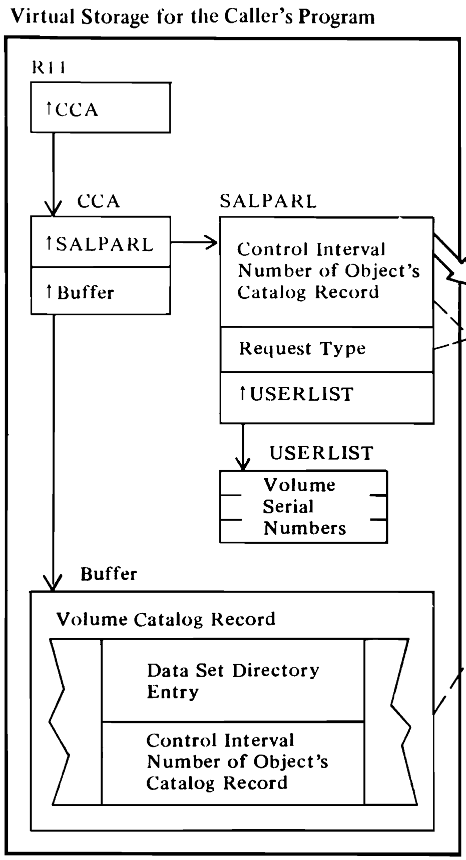
Notes for Diagram DI6

28 IGG0CLB7: IGGPRUS (calls **IGGPMOD (IGG0CLAV)**)

30 IGG0CLB7: IGGPRUS (calls **IGGPGET (IGG0CLBI)**)

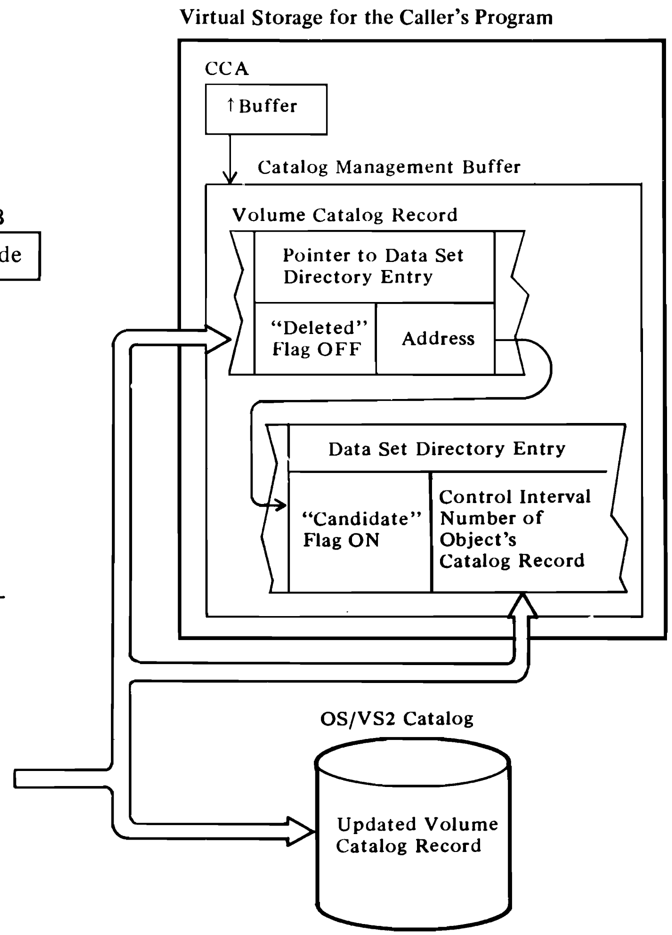
32 IGG0CLB7: IGGPRUS (calls **IGGPGET (IGG0CLBI)**)

Diagram DJ1. SUBALLOCATE: Obtain Additional Space from a Nonunique VSAM Data Space



DI1 4 | ED1 12 | ED2 15

1. Is the volume entry record in the RAB?
   Yes No
2. Was volume entry record found?
   Yes No
3. Return to caller. → CCAPROB Error Code
4. Is the request to assign a candidate volume to the object?
   Yes No → DJ2 9
5. Does the volume catalog record contain a Data Set Directory entry that describes the object?
   No Yes
6. Put "Directory Entry Exists" code in user-provided volume serial list and return. → CCAPROB Return Code
7. Build a Data Set Directory entry to describe the object. (See Diagram DM1, "Modify a Catalog Record Field's Value.")
8. Save return code in user-provided volume serial list and return. → ED2 15



Notes for Diagram DJ1

The Suballocate routine is called to assign a candidate volume to a VSAM object (cluster, data set, index, or catalog) and to assign available space to a VSAM object from one of the data spaces on the caller-specified volume. The caller, either the UPDATE-Extend routine (see Diagram DI1) or the DEFINE CLUSTER routine (see Diagram ED1), builds a list of volume serial numbers to identify each volume to be assigned to the object as a candidate volume. If the caller requests space allocated to the object, the list contains one volume serial number.

1 **IGG0CLAR: IGGPSALL**

The volume entry record may already exist in the RAB, having been put there by the caller of suballocate.

2 **IGG0CLAR: IGGPSALL**

If the volume entry record is not in the RAB, a call is made to IGGPGET (IGG0CLBI) to get the record.

4 **IGG0CLAR: IGGPSALL**

If the request is to assign available space to an object from a specified volume, IGGPSALL calls IGGPSALS (IGG0CLAU).

5 **IGG0CLAR: IGGPSALL**

If the volume catalog record already contains a Data Set Directory Entry set of fields, the volume either is already assigned to the VSAM object as a candidate volume or has some of its space allocated to the VSAM object.

6 **IGG0CLAR: IGGPSALL**

If a Directory Entry already exists for the data set, the return code is set in the user-provided volume serial number list.

7 **IGG0CLAR: IGGPSALL**

IGGPSALL calls IGGPISCI (IGG0CLAG) and IGGPMOD (IGG0CLAV) to add the new data set directory entry to the volume catalog record.

For additional information about topics related to suballocate processing, see:

"Data Areas:"

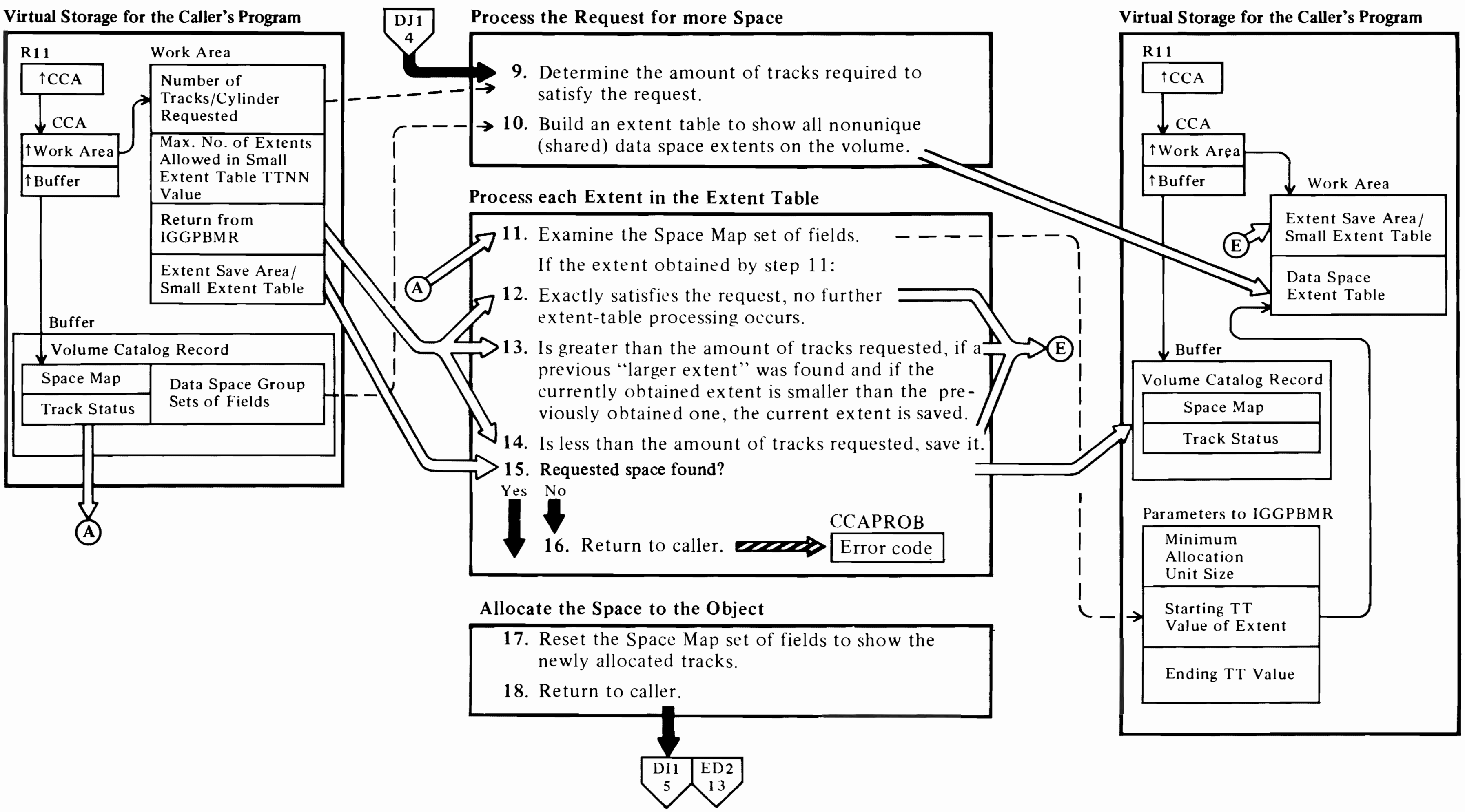
Data Set Directory Entry set of fields description and format

"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DJ2. SUBALLOCATE: Obtain Additional Space from a Nonunique VSAM Data Space



Notes for Diagram DJ2

9 IGG0CLAU: IGGPSALS

If the amount of space requested is a number of cylinders, convert it to a number of tracks.

10 IGG0CLAU: IGGPSALS (calls **IGGPEXT (IGG0CLAZ)**)

The extent table is built by retrieving each extent descriptor (from each Data Space Group set of fields) that might contain enough free space to satisfy the request's minimum allocation requirement (the number of tracks in one control area).

All extents of shared data spaces are described in the table until either there are no more extents to describe or the table is full. If the table is full, step **10** is repeated when steps **11** through **14** are completed, until the extents of all shared data spaces have been examined.

11 IGG0CLAU: IGGPEDS (calls **IGGPBMR (IGG0CLBR)**)

Each extent descriptor in the extent table is in the form:

S# TT NN

where:

S# is the sequence number of the data space's extent.

TT is the extent's starting track number.

NN is the number of tracks in the extent.

The extent descriptors are processed beginning with the lowest TT value in the table, then the next lowest, etc., until all extent descriptors have been processed.

IGGPBMR examines each extent to find an amount of contiguous unallocated tracks at least as large as the request's minimum allocation unit. IGGPBMR examines the Space Map set of fields, starting at bit position (track indicator) TT and ending at bit position (track indicator) TT+NN-1 (usually the extent's track boundaries). If IGGPBMR finds a large enough amount of unallocated tracks, it returns to IGGPEDS with the beginning track number (TT) and the number of tracks (NN). If the data space's extent might contain another amount of unallocated tracks at least as large as the request's minimum allocation unit, IGGPEDS calls IGGPBMR again to examine the rest of the data space's extent.

12 IGG0CLAU: IGGPEDS

If the extent returned by IGGPBMR is the exact number of tracks required to satisfy the caller's request, no further extent table processing is done. Larger or smaller extents obtained from previous extent-table entries are ignored.

13 IGG0CLAU: IGGPEDS

If the extent returned by IGGPBMR is larger than the amount of tracks required to satisfy the request, the extent is saved if either:

- No other "larger-than-requested-amount" extent has been returned yet, or
- The current extent is smaller than a previously obtained "larger-than-requested-amount" extent.

In either case, only one "larger-than-requested-amount" extent value is saved. The "small extent table" (built in step **13**) is ignored and no longer used.

14 IGG0CLAU: IGGPEDS

If the extent returned by IGGPBMR is smaller than the amount of tracks required to satisfy the request, its TTNN value is adjusted so that TT is on a cylinder boundary. If NN is now at least as large as the request's minimum allocation unit (number of tracks for one control area), the extent is saved in the "small extent table" if:

- The table has fewer than five entries (or a caller-specified maximum less than five), or
- The table is full and the current extent's NN value is greater than the table's smallest extent's NN value. The current extent replaces the table's smallest extent.

In either case, the extent is not put in the "small extent table" if it is too small (adjusted NN is less than the minimum allocation unit) or if a "larger-than-requested-amount" extent already exists (see step 12).

If, after all data spaces have been examined, the total of the NN values in the "small extent table" is less than the amount required to satisfy the request, no space is allocated to the object.

17 IGG0CLAU: IGGPSALS (calls **IGGPBMR (IGG0CLBR)**)

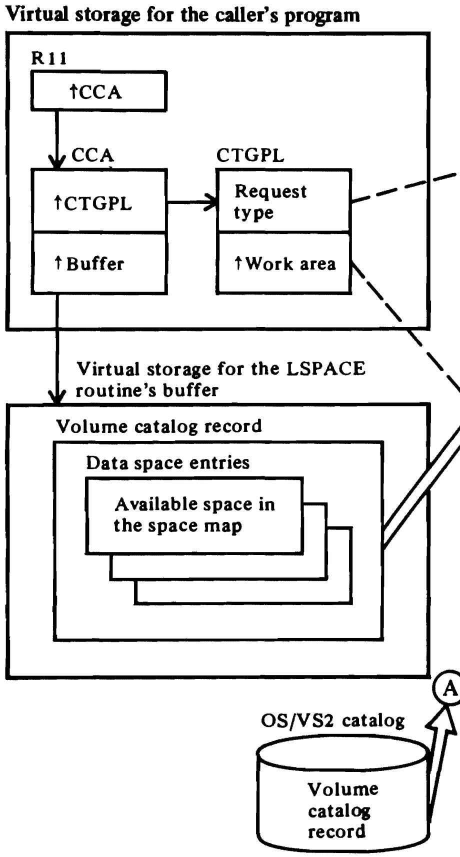
If the selected extent is larger than or equal to the amount of space requested, IGGPBMR adjusts the Space Map set of fields starting at bit position (track indicator) TT, turning off NN bits (NN is the exact number of tracks required to satisfy the request).

If the space is allocated to an object from a number of extents, the "small extent table" is sorted so that the largest NN value is first, the smallest last. IGGPBMR then adjusts the Space Map set of fields for each TTNN value in the "small extent table," until the amount of allocated tracks equals the amount of tracks requested.

18 IGG0CLAU: IGGPSALS

IGGPSALS returns the sequence number of the data space's extent, starting track number, and number of tracks for each extent obtained for the request. The caller uses this information to build extent descriptor entries in the VSAM object's volume information set of fields.

Diagram DK1. LSPACE: Build an "Available Space" Report



DB1 9 | EG2 17

1. Obtain VSAM data space information for all data spaces on the volume. (See Diagram DL1, "Obtain a Catalog Record Field's Value.")
2. Build an available-space report by deriving the available-space information of each data space entry in the volume catalog record.
3. Did the caller provide a work area?

   No / Yes

4. Did an error occur while building the available-space report?

   No / Yes

5. Put a diagnostic message into the caller's work area.

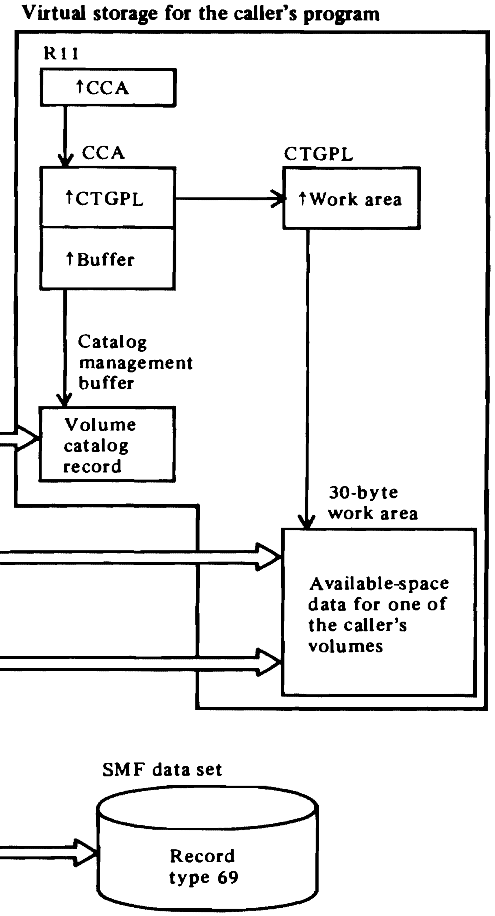
   9

6. Put the available-space report into the caller's work area.
7. Does the OS/VS system include the system management facilities (SMF)?

   No / Yes

8. Write SMF record type 69–VSAM Data Space Information
9. Return to the caller.

CCAPROB: Return code



Notes for Diagram DK1

1 **IGG0CLBK: IGGPLSP** and **IGGPLDCE**

The volume catalog record describes each VSAM data space, and its free space, on the volume.

2 **IGG0CLBK: IGGPLDCS, IGGPLSMS, IGGPLDAS**

Each data space entry derived from the volume catalog record has a field that describes the available space in the data space. The LSPACE routine analyzes each data space entry and calculates the amount of available space in cylinders and tracks. It also records the number of cylinders and tracks in the longest continuous amount of available space.

5 **IGG0CLBK: IGGPLEMP**

A diagnostic message describing the error which occurred during the building of the available space report is placed in the caller's work area.

6 **IGG0CLBK: IGGPLDCE, IGGPLSMP**

If the caller provides a 30-byte work area, the available-space report is put into the work area in the form:

SPACE-CCCC,TTTT,AAAA/cccc,tttt

where:

- CCCC is the number of cylinders of free space in all VSAM data spaces on the volume.
- TTTT is the number of tracks in addition to the number of cylinders of free space on all VSAM data spaces on the volume.
- AAAA is the number of extents of free space in all VSAM data spaces on the volume.
- cccc is the largest number of contiguous freespace cylinders on the volume.
- tttt is the number of contiguous freespace tracks in addition to the cccc value.

8 **IGG0CLBK: IGGPLSP** (calls **IGGPSMFL (IGG0CLBV)**)

See *OS/VS2 System Programming Library: System Management Facilities* for details about SMF record type 69.

9 **IGG0CLBK: IGGPLSP**

For additional information about topics related to LSPACE processing, see:

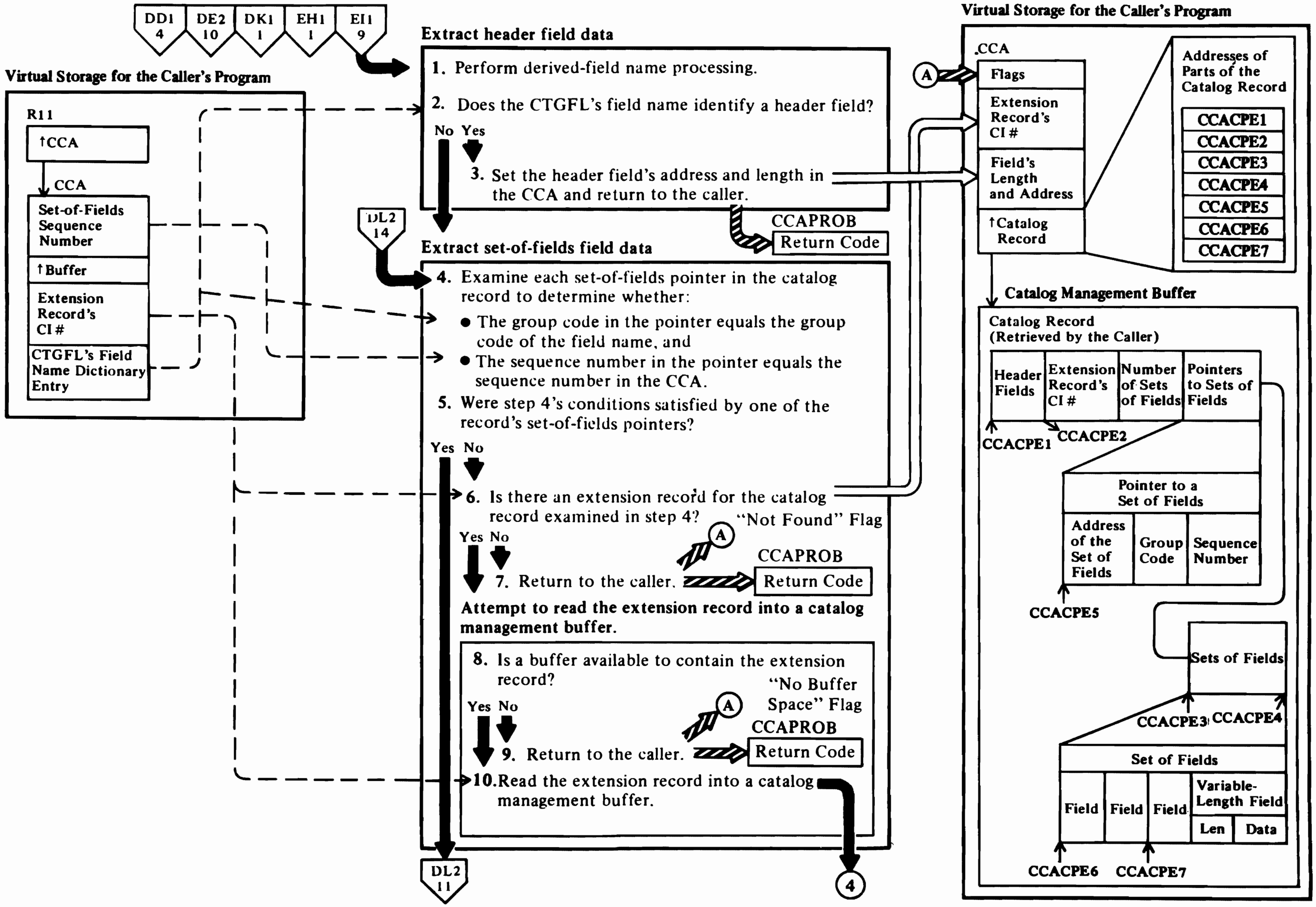
"Data Areas:"

Volume catalog record description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram DL1. Obtain a Catalog Record Field's Value



Notes for Diagram DL1

The Get-Field-Value routine is called by other catalog field management routines to obtain the location and length of a field in a catalog record. The record is in virtual storage in a catalog management buffer. The following results could occur:

- The field is entirely contained in the record in the buffer, and the field's address and length are set in the caller's CCA.
- The field is partially contained in the record in the buffer, and the field's address and partial length are set in the caller's CCA. The CCA also has the "not complete" flag on and contains the control-interval number of the catalog record's extension, which contains more of the field.
- The field is not retrieved because it doesn't exist in the caller-specified set of fields, or because there are no more sets of field in the record, or because no buffer space is available to contain an extension record.

1 IGG0CLBS: IGGPXVAL

If the field name is derived, the field information to be returned does not exist in the physical catalog record but must be generated from the catalog fields, possibly in different catalog records. Derived field names exist only in the volume entry.

2 IGG0CLBA: IGGPLVAL

The field name dictionary is a read-only catalog management table. The catalog field name dictionary contains an entry for each type of catalog record field, based on its field name. The caller puts the dictionary entry identified by the CTGFL's field name into the CCA before calling the get-field-value routine.

If the field name identifies a header field, the field's type code (in the field name dictionary entry) is zero (0). A nonzero type code identifies a set of fields that contains the field identified by the CTGFL's field name.

3 IGG0CLBA: IGGPLVAL

If the field name identifies a header field, and the field is fixed-length, it is at a fixed displacement from the beginning of the catalog record.

The field's address is obtained by adding the displacement in the CTGFL's dictionary entry to the beginning address of the record (the CCA's CCACPE1 value). The field's length is part of CTGFL's dictionary entry.

If the field name identifies a header field, and the field is variable length, the field's address is obtained by using the sequence number in the CTGFL's dictionary entry, which indicates that the field is the first, second, etc., variable-length field.

4 IGG0CLBA: IGGPLVAL

The set of fields pointer (GOP) is used to locate a set of fields. The pointers are grouped together by group code. Within each group of pointers, the pointers are ordered by sequence number.

5 IGG0CLBA: IGGPLVAL

If the caller-specified set of fields pointer (GOP)—identified by its sequence number— is found, its displacement field and flags field specify the location of its set of fields as:

- the number of bytes from the beginning of the record's sets of fields (the CCA's CCACPE3 value plus the set-of-fields pointer's (GOP's) displacement-field value), or
- the control interval number of the extension record that contains the set of fields. The extension record contains a set-of-fields pointer (GOP) that specifies the set of field's location as a number of bytes from the beginning of the record's sets of fields.

6 IGG0CLBA: IGGPGVAL

An extension record might contain the set of fields with the field to be retrieved.

7 IGG0CLBA: IGGPGVAL

8 IGG0CLBA: IGGPGREC

Each catalog record (in a catalog management buffer) is identified by a record area block (RAB) within the CCA. The RAB contains flags that indicate whether or not the buffer can be used to contain another record. If the RAB's "must write" flag is on, the buffer cannot be used for another record until its contents have been written into the catalog.

IGG0CLBA: IGGPGVAL

Each catalog control area (CCA) contains six record area blocks (RABs). Each catalog management request can use a maximum of five buffers. If all buffers are filled and cannot be released, the get-field-value routine sets the CCA's "no buffer space" flag on.

9 IGG0CLBA: IGGPGVAL

10 IGG0CLBA: IGGPGREC

The CCA's "not found" flag is set off before returning to step **4** to examine the extension record's set of fields pointers (GOPs).

For additional information about topics related to field location processing, see:

"Data Areas:"

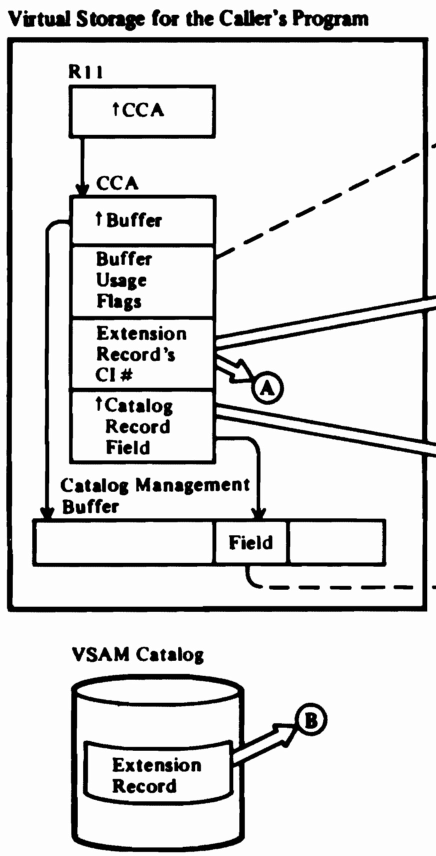
Catalog field names, their descriptions, and the dictionary entry format
Catalog record description and format
Catalog Communications Area (CCA) description and format

"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DL2. Obtain a Catalog Record Field's Value



DL1 5

11. Does the set-of-fields pointer address an extension record?

No Yes

Attempt to read the extension record into a catalog management buffer.

12. Is a buffer available to contain the extension record?

Yes No

13. Return to the caller.

"No Buffer Space" Flag

CCAPROB

Return Code

14. Read the extension record into a catalog management buffer. (See *Virtual Storage Access Method-Extended (VSAM-E) for OS/VS2 MVS: VSAM-E Logic* "GET: Direct Retrieval.")

DL1 4

15. Locate the field in the buffer and set the field's address and length.

16. Is the field variable-length and partially contained in an extension record?

Yes No

17. Return to the caller.

CCAPROB

Return Code

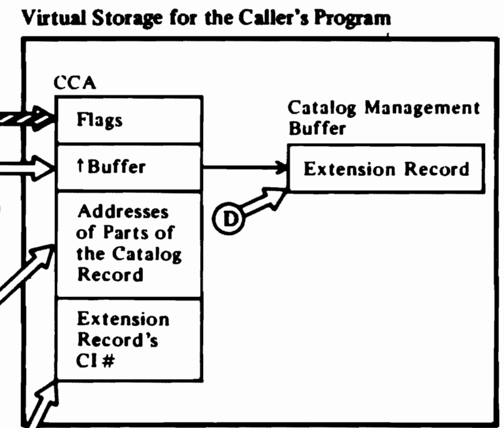
18. Set the extension record's control interval number in the CCA.

19. Return to the caller.

CCAPROB

Return Code

"Not Complete" Flag



Notes for Diagram DL2

11 IGG0CLBA: IGGPLVAL

If the set of fields pointer contains the control interval number (CI#) of an extension record, the set of fields is in that extension record.

12 IGG0CLBA: IGGPGVAL

IGG0CLBA: IGGPGREC

Each catalog record (in a catalog management buffer) is identified by a record area block (RAB) within the CCA. The RAB contains flags that indicate whether or not the buffer can be used to contain another record. If the RAB's "must write" flag is on, the buffer cannot be used for another record until its contents have been written into the catalog.

IGG0CLBA: IGGPGVAL

Each catalog control area (CCA) contains six record area blocks (RABs). Each catalog management request can use a maximum of five buffers. If all buffers are filled and cannot be released, the get-field-value routine sets the CCA's "no buffer space" flag on.

13 IGG0CLBA: IGGPGVAL

14 IGG0CLBA: IGGPGREC

The CCA's "not found" flag is set off before returning to step 4 to examine the extension record's set of fields pointers (GOPs).

15 IGG0CLBA: IGGPLVAL

The field's length is obtained from the CTGFL's dictionary entry (for a fixed-length field) or the first 2 bytes of the field (which are the length bytes of a variable-length field). The field's address is the sum of the address of the set of fields and the displacement in the CTGFL (for a fixed-length field); or is first, second, etc. (as indicated in the CTGFL), variable-length field (for a variable-length field).

16 IGG0CLBA: IGGPLVAL

A variable-length field might be partially contained in an extension record. If so, the field's length is greater than the number of bytes remaining in the record.

17 IGG0CLBA: IGGPLVAL

18 IGG0CLBA: IGGPLVAL

The caller's information requirements might be satisfied with the part of the field that is currently available. If not, the caller (a catalog management routine) returns to the get-field-value routine to obtain the next part of the field from the extension record.

19 IGG0CLBA: IGGPLVAL

The caller can move that part of the field currently in the buffer into a work area. If the rest of the field is required, the caller can return to the get-field-value routine to retrieve the extension record.

For additional information about topics related to field location processing, see:

"Data Areas:"

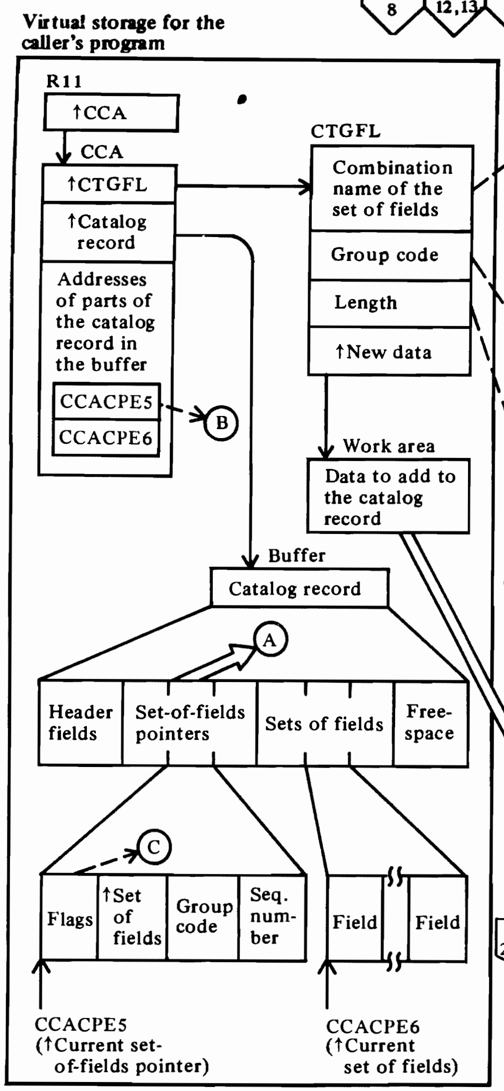
Catalog Communications Area (CCA) description and format

"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DM1. Modify a Catalog Record Field's Value



DH1 8 | DI2 12,13 | ED2 21 | EH1 3 | EH3 23,25

1. Perform derived volume entry field processing.
2. Obtain information about each CTGFL's field name from the field-name dictionary.
3. Perform the type of modification requested:
   - Add a new set of fields to the record → 4
   - Change a field within an existing set of fields → DM2 12
   - Remove a set of fields from the record → DM3 22

4. Examine the set-of-fields pointers. Is there an inactive pointer whose group code equals the new set-of-fields' group code?

   Yes / No

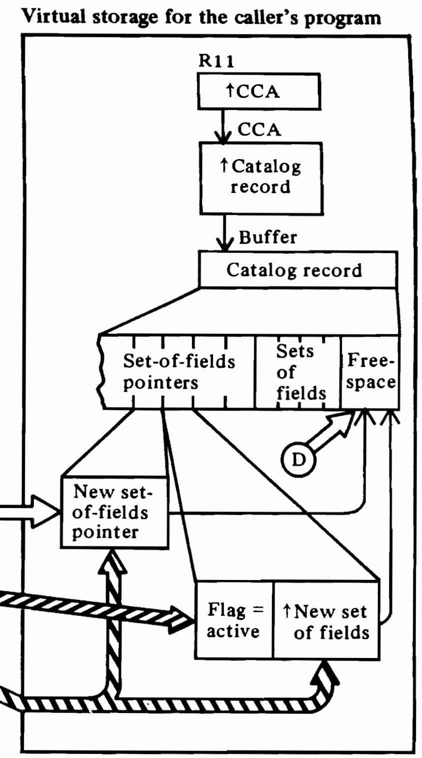
5. Build a new set-of-fields pointer and insert it into the existing set-of-fields pointers.
6. Activate the set-of-fields pointer.
7. Put the displacement of the set of fields from the beginning of the record into the set-of-fields pointer.
8. Initialize all control fields in the new set of fields and put the new set of fields at the end of the catalog record. → D

DM3 23, 26 | DM2 14, 18

9. Move the caller's data into the record, updating existing data in the set of fields. → D
10. Are there more CTGFLs to process?

    Yes / No → DM3 27

11. Process the next CTGFL → 3



Notes for Diagram DM1

1 **IGG0CLAV: IGGPMOD** (calls **IGGPXMOD (IGG0CLBT)**)

For newly-created volume entries, deleted set-of-fields pointers are inserted for the bit map set of fields to ensure that the base volume entry (V) contains the bit map set-of-fields pointers. This bit map set of fields is dynamically added when the first data space set of fields is added.

2 **IGG0CLAV: IGGPMOD** (calls **IGGPSCNC (IGG0CLAY)**)

Each CTGFL is initialized with the dictionary entry associated with the CTGFL's field-name value.

3 **IGG0CLAV: IGGPMOD** (calls **IGGPSFPL (IGG0CLAV)**)

The field parameter list (CTGFL) contains the field's name, type code, length, and displacement from the beginning of the record or set of fields in the case of a fixed-length field. For a variable-length field, contains the field's name, type code, and sequence number. If the field exists, it is either a header field (type code = 0) or a field within a set of fields. If the caller supplied modifying data and test conditions, the field is being altered. If the caller supplied modifying data and no test conditons, a set of fields is to be added to the record. If the caller identified a set of fields combination field name but didn't supply modifying data, the set of fields is being deleted.

4 **IGG0CLAV: IGGPSFPL** (calls **IGGPXDGO (IGG0CLBT)** which in turn calls **IGGPADGO (IGG0CLAW)**)

Every new set of fields is examined by the derived-field processing routine (IGGPXDGO) before being passed on to the normal add field processing routine (IGGPADGO). The derived-field processing routine ensures that certain volume set of fields are never added, added in a different format, or cause dynamic addition of a different set of fields (i.e., bit map set of fields).

5 **IGG0CLAW: IGGPAGOP**

If a new set of fields pointer is built, it is put into the catalog record at the end of its group of set-of-fields pointers. The set-of-fields pointers are grouped by type code with the codes in sequence number order.

If the new set of fields pointer causes the catalog record to overflow, an extension record is obtained from the catalog's free control intervals to free some space in the overflowed catalog record for this new set-of-fields pointer. Some or all sets of fields in the original record are put into the extension record. The set-of-fields pointer's displacement value (in the original record) is replaced with the control interval number of the extension record, for those sets of fields that are moved to an extension record.

In addition, a set-of-fields pointer is built and put into the extension record for each set of fields in the extension record. The set-of-fields pointer in the extension record contains the displacement from the beginning of the sets of fields in the record to its set of fields.

If the new set-of-fields pointer causes the original catalog record to overflow and the catalog record contains only set-of-fields pointers, an extension record is obtained to contain the new set-of-fields pointer (GOP). The original record's extension field contains the control interval number of the extension record.

6 **IGG0CLAW: IGGPAGOP** (calls **IGGPIGOP (IGG0CLAW)**)

The modify routine activates the set of fields pointer by setting it's "inactive" flag off.

7 **IGG0CLAW: IGGPADGO**

8 **IGGOCLAW: IGGPADGO** (calls **IGGPMVGO (IGG0CLAW)**)

The new set of fields might contain fixed-length fields and variable-length fields.

If the new set of fields causes the record to overflow, an extension record is obtained to contain the new set of fields.

9 **IGG0CLAW: IGGPADGO** (calls **IGGPMVGO (IGG0CLAW)**)

Replace the initial field values (from step 7) with the caller-supplied values addressed by the CCA.

10 **IGG0CLAV: IGGPSFPL**

If there are no more CTGFLs to process, IGGPSFPL calls IGGPPREC (IGG0CLAW) to write each updated catalog record into the catalog.

11 **IGG0CLAV: IGGPSFPL**

For additional information about topics related to field modification processing, see:

"Data Areas:"

Catalog record description and format
Set-of-fields descriptions and formats
Set-of-fields pointer description and format

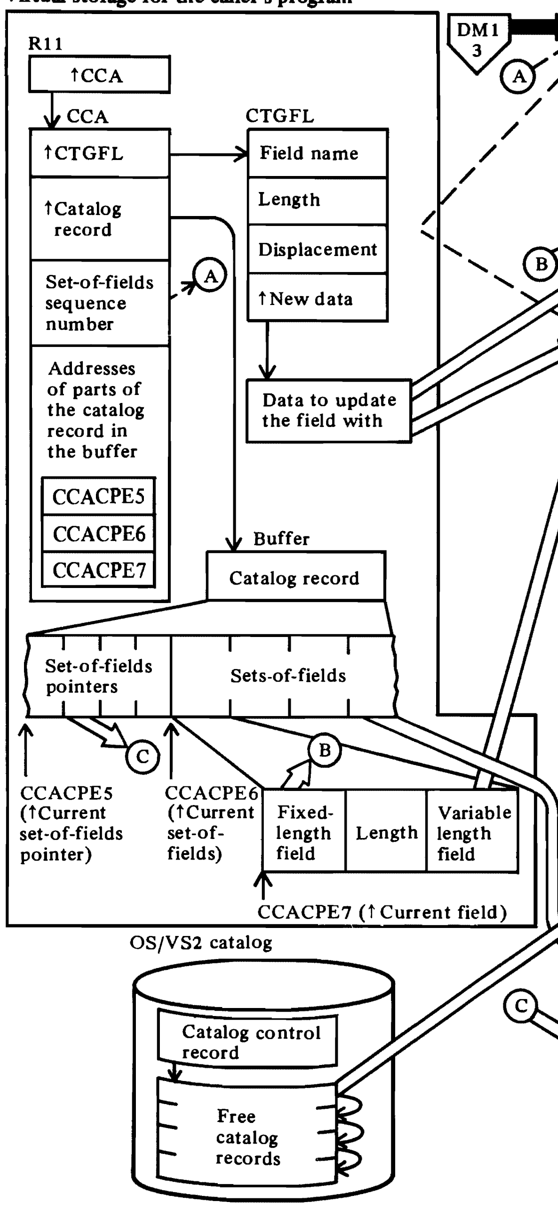
"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DM2. Modify a Catalog Record Field's Value

Virtual storage for the caller's program



DM1 3

Change a field within a set of fields:

12. Obtain the field's length and displacement from the beginning of the record or sequence number (for variable-length fields) and the length and address of the field's updating data.
13. Is a fixed-length field being modified?

No Yes

14. Replace the field's existing data with the updating data.

DM1 10

15. Update the variable-length field's length bytes and data.

If the field's length is:

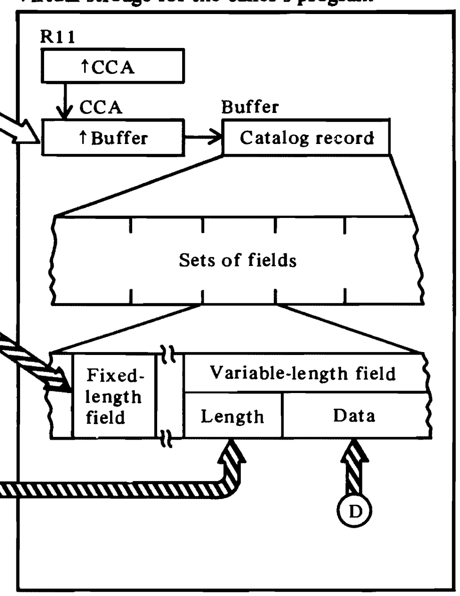
Reduced:

16. Shift the other fields to the left by the number of eliminated bytes.

Increased:

17. Shift the other fields to the right by the number of additional bytes.

Virtual stroage for the caller's program



Obtain an extension record

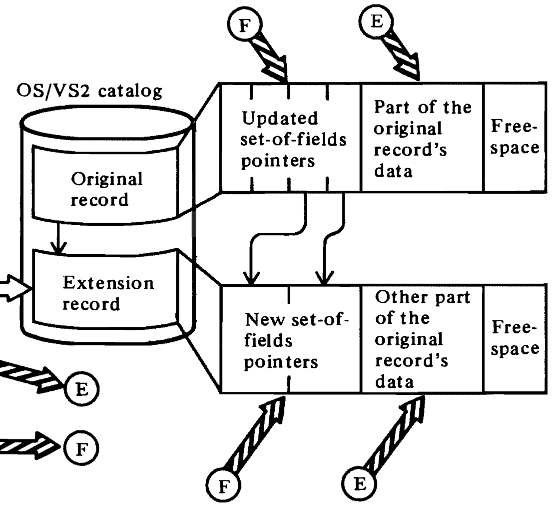
18. Can the record's freespace contain the new or updated set-of-fields' field?

No Yes

DM1 10

19. Assign a free record as an extension to the catalog record.
20. Move the original record's data from the original into the extension until there is enough freespace in the original record for the expansion.
21. Update all set-of-field pointers that are affected by the split.

DM3 27



Notes for Diagram DM2

12 **IGG0CLAV: IGGPSFPL** (calls **IGGPXLT2 (IGG0CLBT)** which in turn calls **IGGPALT2 (IGG0CLAX)**)

Every field to be updated is examined by the derived-field processing routine (IGGPXLT2) before being passed on to the normal update field processing routine (IGGPALT2). The derived-field processing routine ensures that certain volume entry set of fields are never altered, primarily because the altered fields do not physically exist in the catalog records.

IGG0CLAX: IGGPALT2 (calls **IGGPGVAL** (IGG0CLBA))

The CCA's CCACPE7 field contains the field's address. The CTGFL contains the address and length of the data to update the field with.

13 **IGG0CLAX: IGGPALT2**

The CTGFL flags field (from the catalog field name directory) specifies field type.

14 **IGG0CLAX: IGGPALT2**

The CTGFL contains the length and address of the updating data. The data is in the caller's work area.

15 **IGG0CLAX: IGGPMVAR**

The CTGFL flags field (from the catalog field name directory) specifies field type. If the length of the data to update the field with (in the CTGFL) isn't equal to the field's length (in the CCA), the variable-length field's length is either increased or decreased, causing a corresponding reduction or increase in the catalog record's amount of free space.

If the modification occurs in the base catalog record, the affected set of fields is moved from the base catalog record into an extension record.

The variable-length field's length bytes are replaced with the length of the data to update the field with (in the CTGFL).

16 **IGG0CLAX: IGGPSHNK**

The eliminated bytes at the end of the record are added to the record's free space.

17 **IGG0CLAX: IGGPEXPD**

The additional bytes are obtained by reducing the record's free space.

If the increased length causes the catalog record to overflow, an extension record is obtained. The original record's data is split so that part remains in the original record and part is moved into the extension record. Each associated set of fields pointer is updated to show the new position of its set of fields.

19 **IGG0CLAX: IGGPALT2** (calls **IGGPAOCI (IGG0CLAG)**)

The catalog control record (CCR) contains the control-interval number of the first free control interval on the deleted chain. Catalog management allocates the free control interval to the original catalog record as an extension record. The control-interval number of the next free control interval is put into the CCR, and the CCR's free control interval count is decreased by 1.

20 **IGG0CLAX: IGGPALT2**

The set of fields' data is split so that part remains in the original record and part is moved into the extension record.

21 **IGG0CLAX: IGGPALT2**

Each set of fields pointer is updated to show the new position of its set of fields.

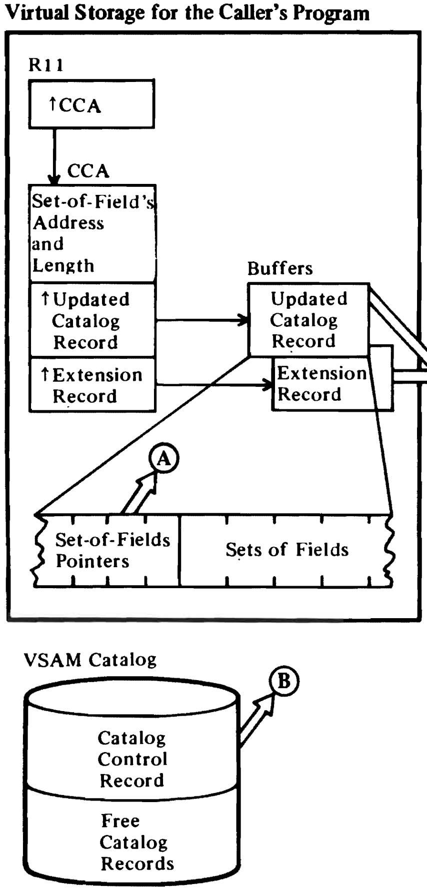
For additional information about topics related to field modification processing, see:

"Data Areas:"

Field Parameter List (CTGFL) description and format

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram DM3. Modify a Catalog Record Field's Value



DM1 3

Delete a set of fields:

22. Deactivate the set-of-fields pointer.

Release an extension record

23. Is the deleted set of fields the only one in an extension record?

Yes No

DM1 10

24. Delete the extension record's control interval number from the record that the extension record extends.

25. Update the extension record to reclassify it as a free catalog record.

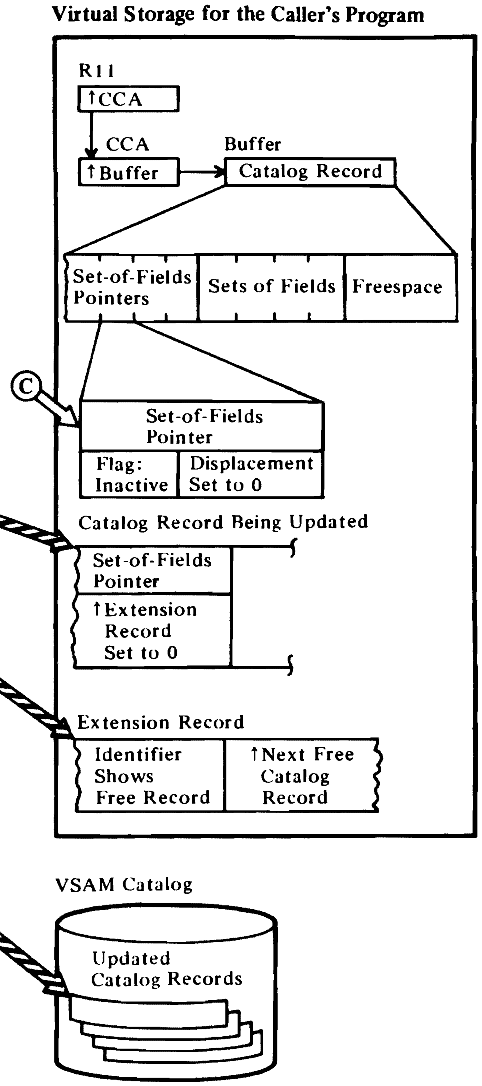
26. Update the catalog control record to show one more free record.

DM1 10 DM1 10 DM2 21

27. Write the updated catalog record from the buffer into the catalog.

28. Return to the caller.

CCAPROB
Return Code



Notes for Diagram DM3

22 IGG0CLAV: IGGSFPL (calls **IGGPXEL2 (IGG0CLBT)** which in turn calls **IGGPDEL2 (IGG0CLAV)**)

Every set of fields to be deleted is examined by the derived field processing routine (IGGPXEL2) before being passed on to the normal delete set of fields processing routine (IGGPDEL2). The derived field processing routine ensures that certain nonexistent sets of fields are not deleted and that the bit map set of fields is updated.

IGG0CLAV: IGGPDEL2

The set of fields pointer's "inactive" flag is set on, thereby deactivating it. The set of fields pointer's type code and sequence number fields are unchanged. The set of fields is removed from the record and the displacement field is set to 0, if the set of fields is in the same record as the set-of-fields pointer; otherwise, the following is done:

23 IGG0CLAV: IGGPDEL2

If the extension record contains no data after the set of fields is removed, the record is reclaimed by catalog management as a free control interval.

25 IGG0CLAV: IGGPDEL2

26 IGG0CLAV: IGGPDEL2

The catalog contains "once used but now free" control intervals. These are control intervals that have been deleted. The catalog control record (CCR) contains the count of deleted control intervals available to the catalog and the control interval number of the first deleted control interval. All deleted control intervals are chained together in the catalog. When a control interval is added to the deleted control interval chain, catalog management puts the CCR's deleted control interval number into the new deleted control interval and puts the new deleted control interval's control-interval number into the CCR. The CCR's deleted-control-interval count is increased by 1.

If a deleted variable-length field in a set of fields spans several extension records, those extension records are deleted also.

27 IGG0CLAV: IGGPMOD

28

The modify routine returns to the caller (a catalog management routine) when the catalog record fields are updated.

For additional information about topics related to field modification processing, see:

"Data Areas:"

Catalog record description and format
Extension catalog record description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram EA1. Catalog Management Services Table of Contents

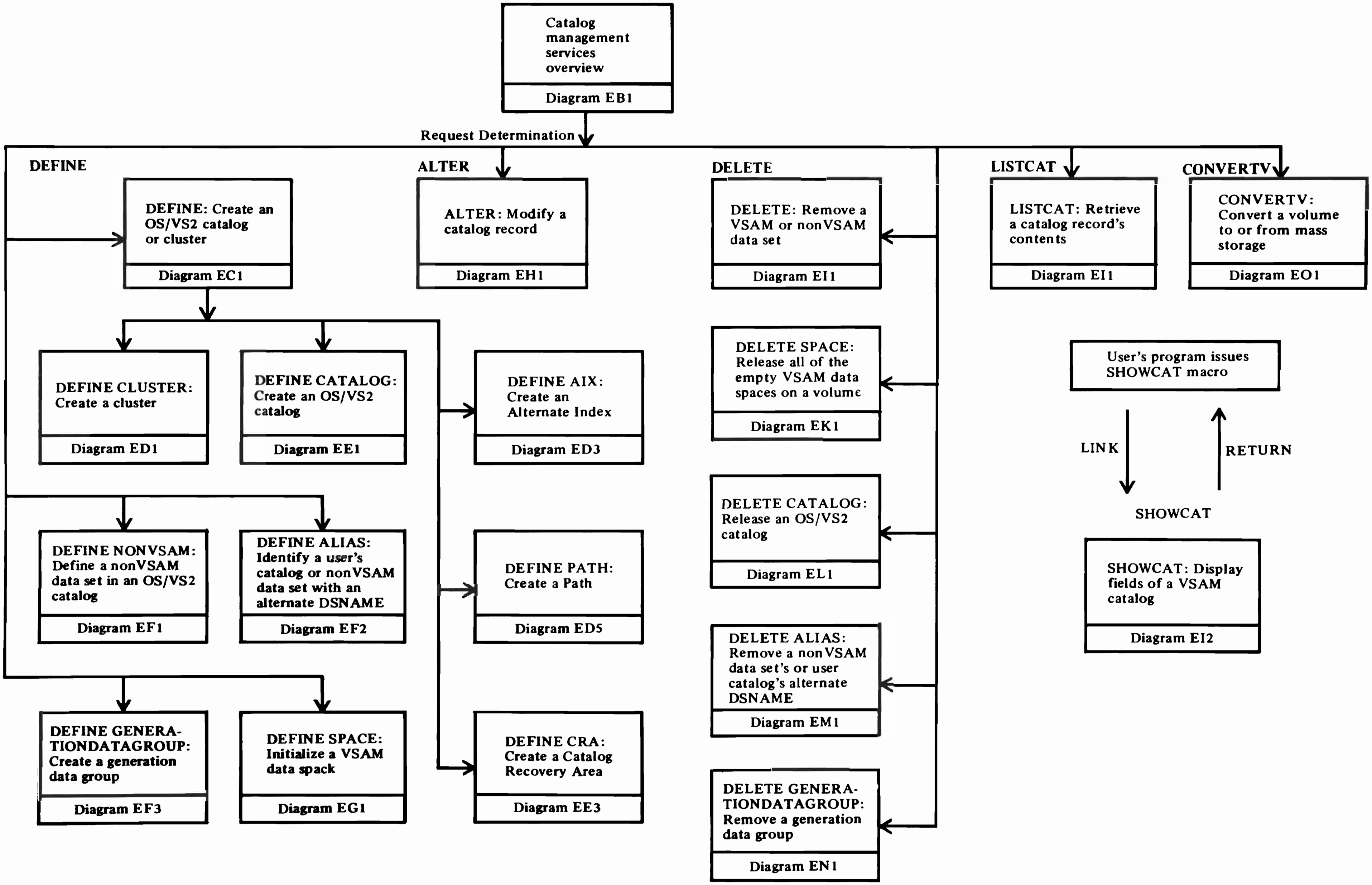
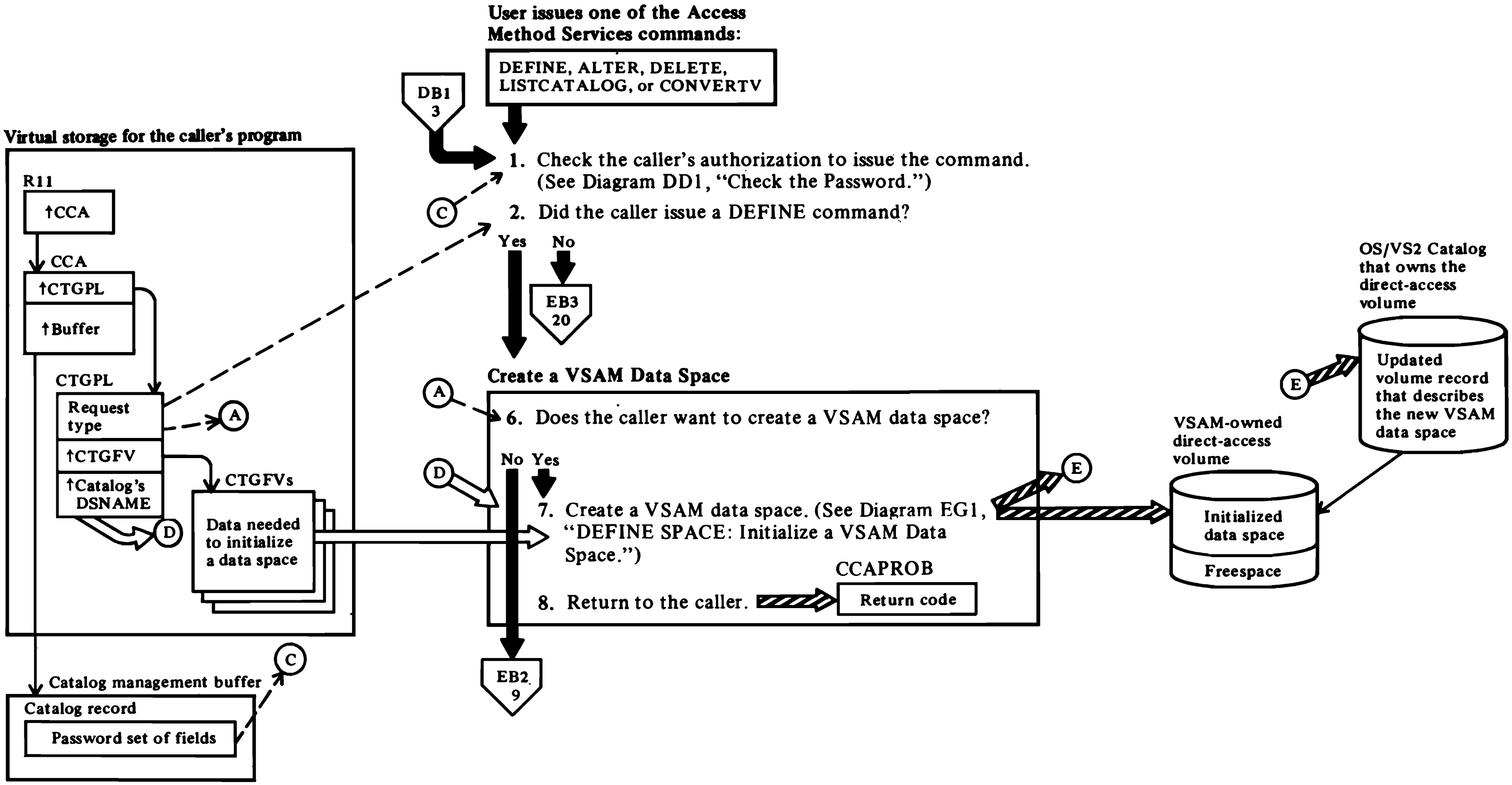


Diagram EB1. Catalog Management Services Overview



Notes for Diagram EB1

Catalog Management Services is a group of OS/VS2 catalog management modules that respond to Access Method Services commands. The Catalog Management Services Driver routine, IGGPCDVR (IGG0CLAT), calls other Catalog Management Services routines as described in the table below.

For additional information about topics related to Catalog Management Services—overview — processing, see:

"Data Areas:"

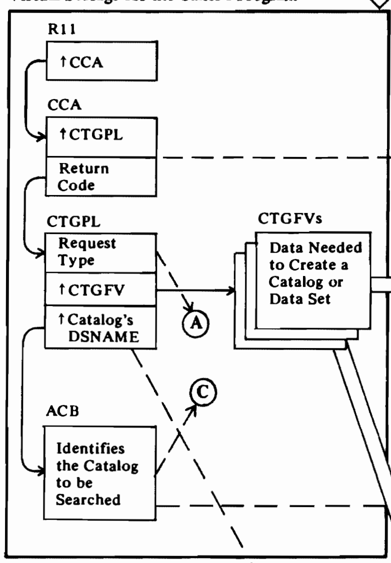
Access method control block (ACB) description and format

"Diagnostic Aids:"

Catalog management return codes

Access Method Services Command	**IGG0CLAT Calls**	**Related Method of Operation Diagram**
DEFINE CLUSTER, AIX, PATH, PAGESPACE, and DEFINE USERCATALOG	IGG0CLAL	EC1
DEFINE CRA	IGG0CLB4	EE3
DEFINE NONVSAM	IGG0CLBH	EF1
DEFINE ALIAS	IGG0CLAO	EF2
DEFINE GENERATIONDATAGROUP	IGG0CLA4	EF3
DEFINE SPACE	IGG0CLAQ	EG1
ALTER	IGG0CLBD	EH1
LISTCAT	IGG0CLBQ	EI1
DELETE ALIAS	IGG0CLA5	EM1
DELETE CLUSTER, AIX, PATH, PAGESPACE, and DELETE NONVSAM	IGG0CLBG	EJ1
DELETE GENERATIONDATAGROUP	IGG0CLA5	EN1
DELETE SPACE	IGG0CLBL	EK1
DELETE USERCATALOG	IGG0CLAF	EL1
CONVERTV	IGG0CLBZ	EO1

Diagram EB2. Catalog Management Services Overview



Create an OS/VS2 User Catalog

EB1 3

6. Does the caller want to create an OS/VS2 user catalog?

No Yes

7. Search the master catalog to ensure that the new catalog's DSNAME doesn't duplicate an existing DSNAME. (See Diagram DC1, "Search: Retrieve a Catalog Record.")
8. Did the search routine set the no-record-found error code?

Yes No

9. Return to the caller. CCAPROB Error Code
10. Build the OS/VS2 user catalog. (See Diagram EC1. "DEFINE: Create a Catalog or Cluster.")
11. Return to the caller. CCAPROB Error Code

DB1 10

Create a Data Set

12. Search the catalog specified in the caller's CTGPL to ensure that the new data set's DSNAME doesn't duplicate an existing DSNAME. (See Diagram DC1, "Search: Retrieve a Catalog Record.")
13. Did the search routine set the no-record-found error code?

Yes No

14. Return to the caller. CCAPROB Error Code
15. Process the DEFINE command to create:
   - A cluster (a VSAM data set, with or without an index). (See Diagram EC1, "DEFINE: Create a Catalog, AIX, Path or, Cluster.")
   - A nonVSAM data set. (See Diagram EF1. "DEFINE NONVSAM: Define a NonVSAM Data Set in a catalog.")
   - A catalog recovery area. (See Diagram EE3, "DEFINE CRA: Create a Catalog Recovery Area.")
   - An alternate name for a user's catalog or nonVSAM data set (See Figure EF2, "DEFINE ALIAS: Identify a User's Catalog or NonVSAM Data Set with an Alternate DSNAME.")
   - A generation data group (See Figure EF3, "DEFINE GENERATIONDATAGROUP: Create a Generation Data Group.")
16. Return to caller. CCAPROB Return Code

DB1 10

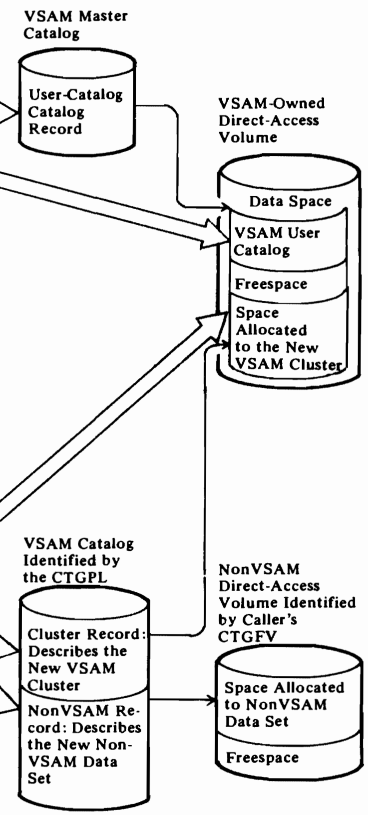
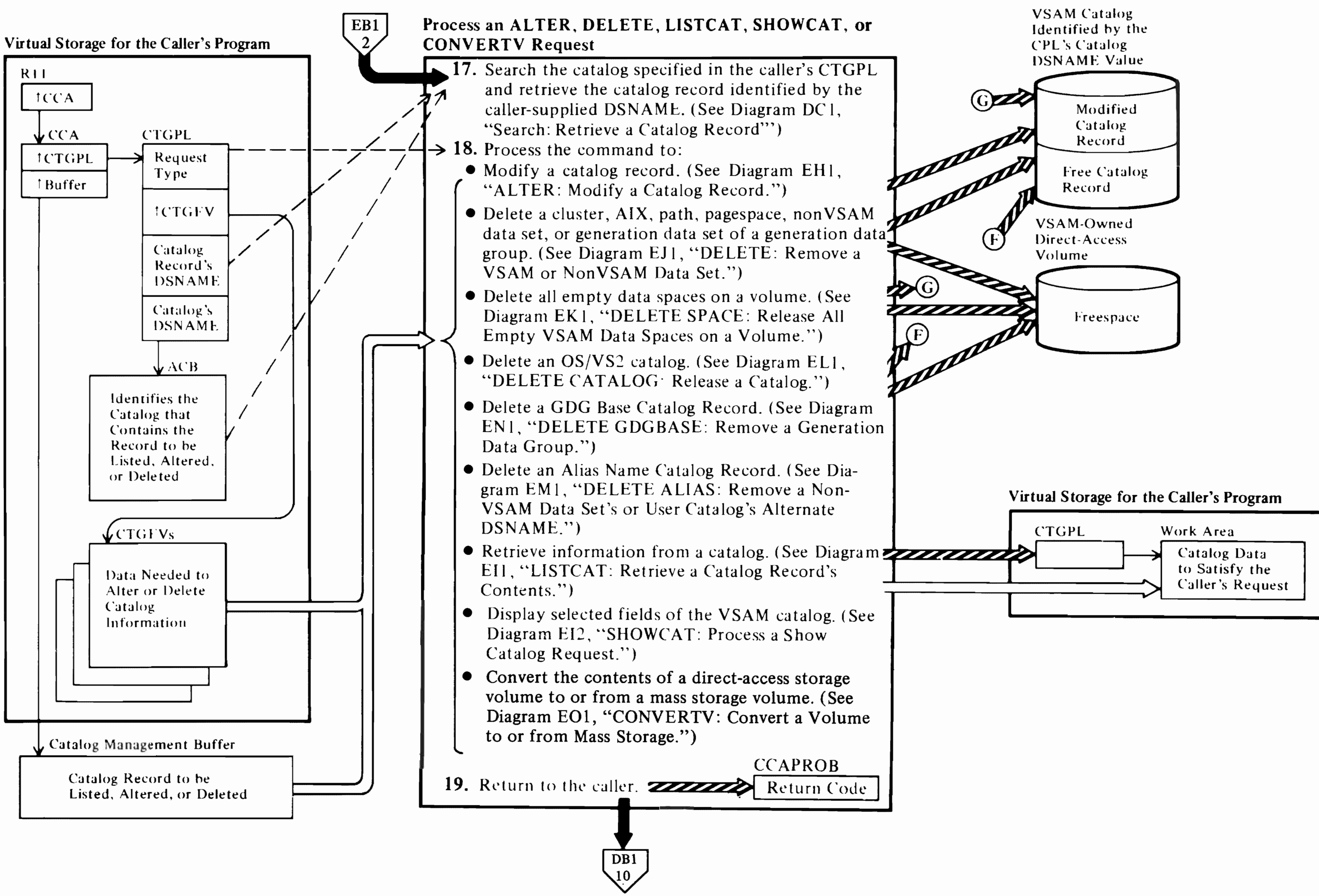


Diagram EB3. Catalog Management Services Overview



Process an ALTER, DELETE, LISTCAT, SHOWCAT, or CONVERTV Request

17. Search the catalog specified in the caller's CTGPL and retrieve the catalog record identified by the caller-supplied DSNAME. (See Diagram DC1, "Search: Retrieve a Catalog Record")

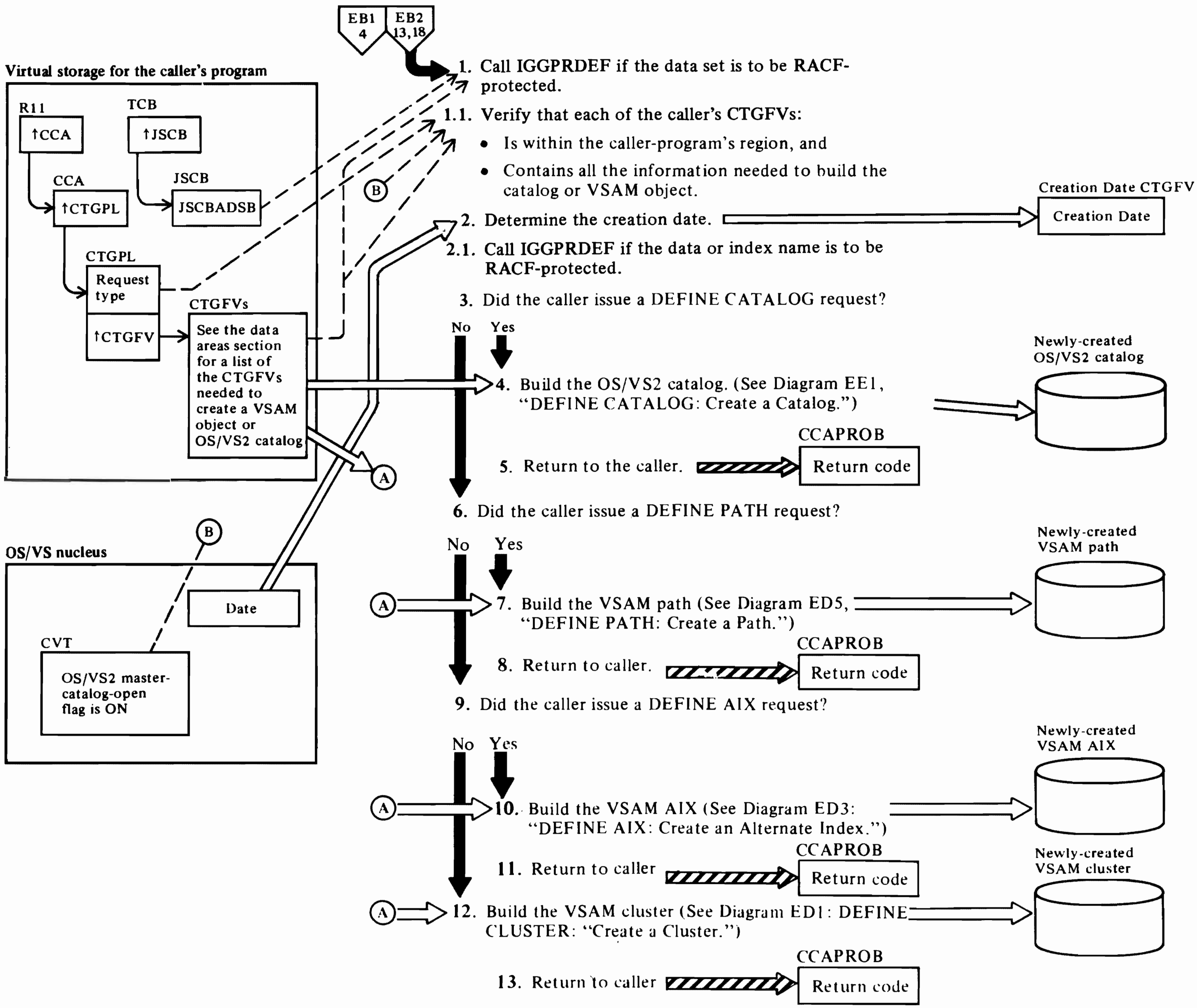
18. Process the command to:
- Modify a catalog record. (See Diagram EH1, "ALTER: Modify a Catalog Record.")
- Delete a cluster, AIX, path, pagespace, nonVSAM data set, or generation data set of a generation data group. (See Diagram EJ1, "DELETE: Remove a VSAM or NonVSAM Data Set.")
- Delete all empty data spaces on a volume. (See Diagram EK1, "DELETE SPACE: Release All Empty VSAM Data Spaces on a Volume.")
- Delete an OS/VS2 catalog. (See Diagram EL1, "DELETE CATALOG: Release a Catalog.")
- Delete a GDG Base Catalog Record. (See Diagram EN1, "DELETE GDGBASE: Remove a Generation Data Group.")
- Delete an Alias Name Catalog Record. (See Diagram EM1, "DELETE ALIAS: Remove a Non-VSAM Data Set's or User Catalog's Alternate DSNAME.")
- Retrieve information from a catalog. (See Diagram EI1, "LISTCAT: Retrieve a Catalog Record's Contents.")
- Display selected fields of the VSAM catalog. (See Diagram EI2, "SHOWCAT: Process a Show Catalog Request.")
- Convert the contents of a direct-access storage volume to or from a mass storage volume. (See Diagram EO1, "CONVERTV: Convert a Volume to or from Mass Storage.")

19. Return to the caller.

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram EC1. DEFINE: Create an OS/VS2 Catalog, VSAM Cluster, VSAM AIX, or VSAM Path



Notes for Diagram EC1

When the user issues the Access Method Services DEFINE command to create a catalog (either user or master) or a VSAM object, the catalog management services DEFINE: Initial Processing modules ensure that the caller provided all the information necessary to create a catalog or a VSAM object.

1 IGG0CLAT: IGGPCDVR

This is done just before calling IGG0CLAL. IGGPRDEF will process all user-specified entity names, issuing RACDEF SVCs (133) to define these entities to the RACF subsystem.

1.1 CTGFVs are checked by internal procedures in the IGG0CLAL and IGG0CLAN modules, as shown in the table below.

2 IGG0CLAL: IGGPDEDE

2.1 IGG0CLAL: IGGPDEDE

This is done for the user-supplied name or the generated name.

4 IGG0CLAN: IGGPDSCB (calls **IGGPDCDA (IGG0CLAP)** and **IGGPDEFC (IGG0CLAS)**)

7 IGG0CLAL: IGGPDDEP

10 IGG0CLAL: IGGPDCWC

12 IGG0CLAN: IGGPDSCB (calls **IGGPDRDA (IGG0CLAN)** and **IGGPDBDI (IGG0CLAJ)**)

CTGFV Name	Module	Procedure
Cluster CTGFV	IGG0CLAL	IGGPDCWC
AIX CTGFV	IGG0CLAL	IGGPDCWC
Caller's work area	IGG0CLAL	IGGPDCWC
Data CTGFV structure	IGGOCLAL	IGGPDFSC
Index CTGFV structure	IGG0CLAL	IGGPDFSC
Data dsname CTGFV	IGG0CLAL	IGGPDDNP
Index dsname CTGFV	IGG0CLAL	IGGPDDNP
Path CTGFV	IGG0CLAL	IGGPDDEP
Data space CTGFV	IGG0CLAL	IGGPDBVC
Catalog's space CTGFV	IGG0CLAL	IGGPDCSF
Key range CTGFV	IGG0CLAL	IGGPDRPG
Space parameter CTGFLs	IGG0CLAN	IGGPDSPF
Buffer size CTGFLs	IGG0CLAN	IGGPDBSF
Average record size CTGFLs	IGG0CLAN	IGGPDALR

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram ED1. DEFINE CLUSTER: Create a Cluster

Virtual storage for the caller's program

R11
↑CCA

CCA
↑CTGPL
↑Catalog's ACB
↑Buffer

CTGPL
↑CTGFVs

CTGFVs
↑CTGFLs

CTGFLs
Describes data set and index characteristics

CTGFLs
Describes data set and index space requirements

CTGFLs
Describes the passwords

CTGFLs
Describes data set and index volume usage and serial numbers

Catalog management buffers
Free catalog record #1
Free catalog record #2
Free catalog record #3

EC1 12

1. Build the cluster's AMDSB set of fields.
2. Compute the amount of direct-access space required for the cluster.
3. Obtain contiguous free catalog records from the catalog specified in the caller's CTGPL and assign them to the cluster.

Build the cluster catalog record

4. Initialize the cluster catalog record's header fields.
5. Build an association set of fields to point to the data set catalog record and, if the cluster is key-sequenced, the index catalog record.
6. Build the cluster's password set of fields.
7. Compute the cluster catalog record's size.
8. Write the cluster catalog record into the catalog.

Do steps 9 through 23 to:

- Build the data set catalog record to describe the data set and its assigned space.
- Build the index catalog record to describe the data set's index and its assigned space, if the cluster is key-sequenced.

9. Initialize the catalog record's header fields.

Obtain direct-access space to satisfy the data set's or index's space allocation requirements

10. Can the data set or index share its space with other VSAM data sets and indexes?

Yes No

11. Allocate a VSAM data space to the data set or index. (See Diagram EG1, "DEFINE SPACE: Initialize a VSAM Data Space.")
12. Allocate space to the data set or index from an existing VSAM data space. (See Diagram DJ1, "Suballocate: Obtain Additional Space from a Nonunique VSAM Data Space.")

ED2 13

Catalog management work area
AMDSB set-of-fields
Amount of space needed for the cluster

Catalog management buffers
Cluster catalog record
Data set catalog record
Index catalog record

OS/VS2 catalog identified by the caller's CTGPL
New cluster catalog record

VSAM-owned direct-access volume
Space allocated to the data set or index

Notes for Diagram ED1

This figure describes the processes performed by catalog management services routines when the user issues the Access Method Services DEFINE CLUSTER command in the form:

DEFINE
CLUSTER
(parameterlist)
[CATALOG (catname/password)]

where:

- *catname* is the name of the catalog that will contain the cluster, data set, and index catalog records that describe the user's data set.
- *password* is the catalog's master, control interval, or update password, if the catalog is protected by passwords.
- *parameterlist* is a list of optional and required parameters that define the cluster's characteristics.

See *OS/VS2 Access Method Services* for details about the DEFINE command parameters.

Note: When an AMS user issues the DEFINE PAGESPACE command, the user wants to identify a single-extent amount of direct-access device space called a pagespace. The user can then create data sets that reside in the pagespace. Catalog management services processes a DEFINE PAGESPACE request by creating an entry-sequenced data set with fixed-length control intervals and track overflow. The pagespace catalog record is similar to an entry-sequenced data set's record, except that the record identifier is "P".

1 IGG0CLAN: IGGPDRDA

The AMDSB contains the cluster's statistics and fixed characteristics. The AMDSB set of fields is in the Data catalog record (for an entry-sequenced data set) and in the Data and Index catalog records (for a key-sequenced data set). Each time the cluster is opened, the AMDSB is retrieved from the data set catalog record. When the cluster is closed, the AMDSB is updated and rewritten into the data set catalog record.

2 IGG0CLBX: IGGPDSPC

The field vector table (CTGFV) contains addresses of buffer-size and record-length field parameter lists (CTGFLs). This data is used to determine the data set's control-interval and control-area size. If the data set is key-sequenced, other buffer-size and record-length CTGFLs determine the index's control-interval and control-area size. If the key-sequenced data set is divided into key ranges, the size of each key range is determined.

IGG0CLBX

IGG0CLBX determines the amount of secondary storage needed. For Pagespaces, IGG0CLBX calculates this space using track overflow where appropriate.

IGG0CLBY: IGGPDRSP

IGGPDRSP is called to calculate the space parameters.

3 IGG0CLAN: IGGPDCCE

A user's data set is described by a cluster catalog record, a data set catalog record, and, if the data set is key-sequenced, an index catalog record.

IGG0CLAG: IGGPAOCI

IGGPAOCI is called to obtain two (for an entry-sequenced cluster) or three (for a key-sequenced cluster) contiguous catalog control intervals to contain the cluster, data set, and (if the cluster is key-sequenced) index catalog records.

8 IGG0CLAN: IGGPDCCE

The DEFINE routine issues an ADDREC macro instruction to write the cluster record into the catalog.

9 IGG0CLAJ: IGGPDBDI

10 IGG0CLAJ: IGGPDSPO

11 IGG0CLAJ: IGGPDSPO (calls **IGGPDEFS (IGG0CLAQ)**)

A Data Space Group set of fields is added to the volume catalog record, and the data set's name is added to the volume catalog record's data set directory.

12 IGG0CLAJ: IGGPDSPO (calls **IGGPSALL (IGG0CLAR)**)

The data set's name is added to the volume catalog record's Data Set Directory Entry set of fields, and the volume catalog record's Data Space Group set of fields is updated to show the cylinders and tracks allocated to the new data set or index.

For additional information about topics related to DEFINE CLUSTER processing, see:

"Data Areas:"

Catalog record description and format
Data set catalog record description and format
Volume catalog record description and format
Data set group set of fields description
Data set directory entry set of fields description
Access Method data set statistics block (AMDSB) description and format

Diagram ED2. DEFINE CLUSTER: Create a Cluster

Virtual Storage for the Caller's Program

R11
↑CCA

CCA
↑Work Area
↑Buffers
↑CTGPL

CTGPL
↑CTGFV

CTGFVs
↑CTGFLs

Work Area
- Low and High CCHH Values for Each Extent Allocated to the Data Set, Index, or Data Set's Key Range
- Extent Descriptors (From Steps 13 & 14)
- Volume Information Sets of Fields (From Steps 16, 17, and 18)

CTGFLs
- Describes Data Set and Index Volume Usage and Serial Numbers

CTGFLs
- Describes Data Set and Index Characteristics

Catalog Management Buffers
- Data Set Catalog Record
- Index Catalog Record

ED1 12

Assign space (continued)

13. Build an extent descriptor that describes each extent allocated to the data set or index.
14. If the data set is divided into key ranges, assign space to each of the data set's key ranges.
15. Assign candidate volumes to the data set or index. (See Diagram DJ1, "SUBALLOCATE" – Assign a Candidate Volume to the Object.)

Build a volume information set of fields to describe:

16. All extents allocated to the data set or index on each volume.
17. All extents allocated to each of the data set's key ranges on each volume.
18. Each of the data set's or index's candidate volumes.
19. Is the "sequence set with data" option specified?
No Yes
20. Assign part of each of the data set's extents to the sequence set.
21. Put each volume information set of fields into the data set or index catalog record. (See Diagram DM1, "Modify a Catalog Record Field's Value.")
22. Write the data set or index catalog record into the catalog.
23. Free the virtual storage of the work areas obtained by the Define routine.

Write SMF record type 63

24. Does the OS/VS system include the system management facility (SMF)?
No Yes
25. Write SMF record type 63 – Cluster Cataloged.
26. Return to the caller.

CCAPROB
- Return Code

Virtual Storage for the Caller's Program

R11
↑CCA

CCA
↑Work Area
↑Buffers

Work Area
- Extent Descriptors
- Volume Information Sets of Fields

Catalog Management Buffers
- Updated Data Set Catalog Record
- Updated Index Catalog Record

VSAM Catalog Identified by the Caller's CTGPL
- New Data Set Catalog Record
- New Index Catalog Record

SMF Data Set
- Record Type 63

Notes for Diagram ED2

13 IGG0CLAJ: IGGPDSEX

14 IGG0CLAJ: IGGPDSPO

Each key range is assigned physical space and space allocation continues until all key ranges have been assigned space.

15 IGG0CLAJ: IGGPDCNV

A candidate volume is available to contain part of the cluster if more space is needed later. None of the candidate volume's space is allocated to the data set or index when the cluster is created.

16 IGG0CLAK: IGGPDBVO

Each volume that contains a part of the data set or index is described by a volume information set of fields within the data set or index catalog record.

17 IGG0CLAK: IGGPDRNG

If the data set is divided into key ranges, each key range's space on a volume is described in a separate volume information set of fields. If the key range's space is on more than one volume, each volume that contains part of the key range's space is described in a separate volume information set of fields.

18 IGG0CLAK: IGGPDBCV

IGGPDBCV builds a volume information set of fields for each candidate volume of the data set.

20 IGG0CLAK: IGGPDSSP

If the "sequence set with data" option is specified, part of the data set's space is allocated to the index for sequence set records. The low and high CCHH values in the index record's volume information set of fields are those of the extent obtained for the data set. The low and high RBA values are for the sequence set and are relative to the index addresses.

21 IGG0CLAK: IGGPDMOP

IGGPDMOP calls IGGPMOD (IGG0CLAV) to add each volume information set of fields to the record.

22 IGG0CLAK: IGGPDMOP

A catalog management routine writes the completed data set or index catalog record into the VSAM catalog and frees the catalog management buffer that contains the record.

23 IGG0CLA8: IGGPDFRS

IGGPDFRS frees all unneeded storage resources.

25 IGG0CLAJ: IGGPDBDI (calls **IGGPSMFA (IGG0CLBV)**)

See *OS/VS2 System Programming Library: System Management Facilities* for details of SMF record type 63—VSAM Data Set Cataloged. Record type 63 is written after a VSAM cluster is defined and whenever the definition is altered.

For additional information about topics related to DEFINE CLUSTER processing, see:

"Data Areas:"

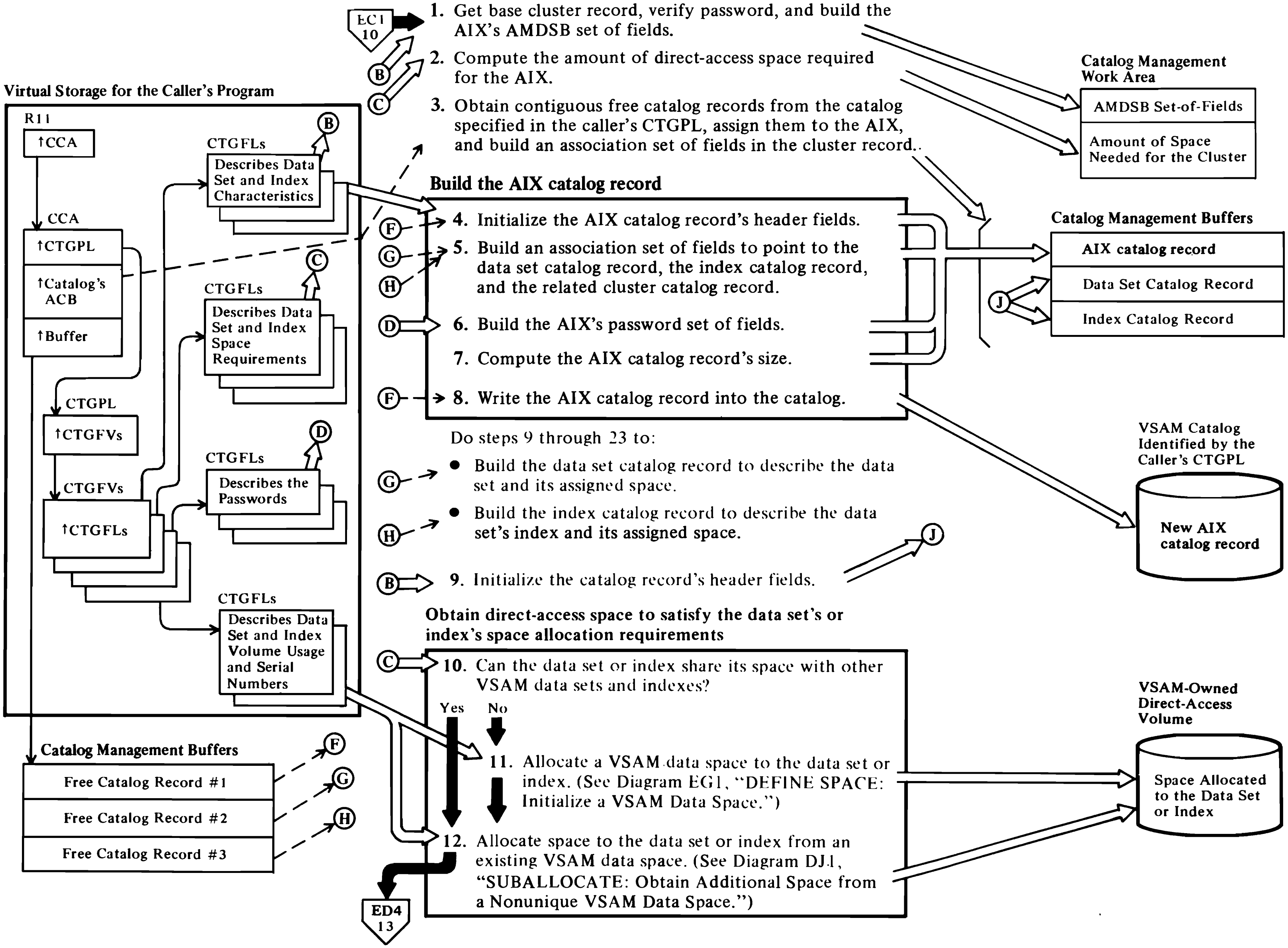
Data set catalog record description and format
Index catalog record description and format
Volume information set of fields description

"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram ED3. DEFINE AIX: Create an Alternate Index



Notes for Diagram ED3

This figure describes the processes performed by catalog management services routines when the user issues the Access Method Services DEFINE AIX command in the form:

DEFINE
AIX
(parameterlist)
[CATALOG (catname/password)]

where:

- *catname* is the name of the catalog that will contain the AIX, data set, and index catalog records that describe the user's data set.
- *password* is the catalog's master, control interval, or update password, if the catalog is protected by passwords.
- *parameterlist* is a list of optional and required parameters that define the AIX's characteristics.

See *OS/VS2 Access Method Services* for details about the DEFINE command parameters.

Note: When an AMS user issues the DEFINE PAGESPACE command, the user wants to identify a single-extent amount of direct-access device space called a pagespace. The user can then create data sets that reside in the pagespace. Catalog management services processes a DEFINE PAGESPACE request by creating an entry-sequenced data set with fixed-length control intervals and track overflow. The pagespace catalog record is similar to an entry-sequenced data set's record, except that the record identifier is "P".

IGG0CLB9: IGGPPRPW

1 IGG0CLAN: IGGPDRDA

The AMDSB contains the cluster's statistics and fixed characteristics. The AMDSB set of fields is in the Data catalog record (for an entry-sequenced data set) and in the Data and Index catalog records (for a key-sequenced data set). Each time the cluster is opened, the AMDSB is retrieved from the data set catalog record. When the cluster is closed, the AMDSB is updated and rewritten into the data set catalog record.

2 IGG0CLAN: IGGPDSPC

The field vector table (CTGFV) contains addresses of buffer-size and record-length field parameter lists (CTGFLs). This data is used to determine the data set's control-interval and control-area size. If the data set is key-sequenced, other buffer-size and record-length CTGFLs determine the index's control-interval and control-area size. If the key-sequenced data set is divided into key ranges, the size of each key range is determined.

IGG0CLBX

IGG0CLBX determines the amount of secondary storage needed. For Pagespaces, IGG0CLBX calculates this space using track overflow where appropriate.

IGG0CLBY: IGGPDRSP

IGGPDRSP is called to calculate the space parameters.

3 IGG0CLAG: IGGPAOCI

IGG0CLB9: IGGPMODC

IGGPAOCI is called to obtain three contiguous catalog control intervals to contain the AIX, data set, and index catalog records.

8 IGG0CLAN: IGGPDCCE

The DEFINE routine issues an ADDREC macro instruction to write the AIX record into the catalog.

9 IGG0CLAJ: IGGPDBDI

10 IGG0CLAJ: IGGPDSPO

11 IGG0CLAJ: IGGPDSPO (calls **IGGPDEFS (IGG0CLAQ)**)

A Data Space Group set of fields is added to the volume catalog record, and the data set's name is added to the volume catalog record's data set directory.

12 IGG0CLAJ: IGGPDSPO (calls **IGGPSALL (IGG0CLAR)**)

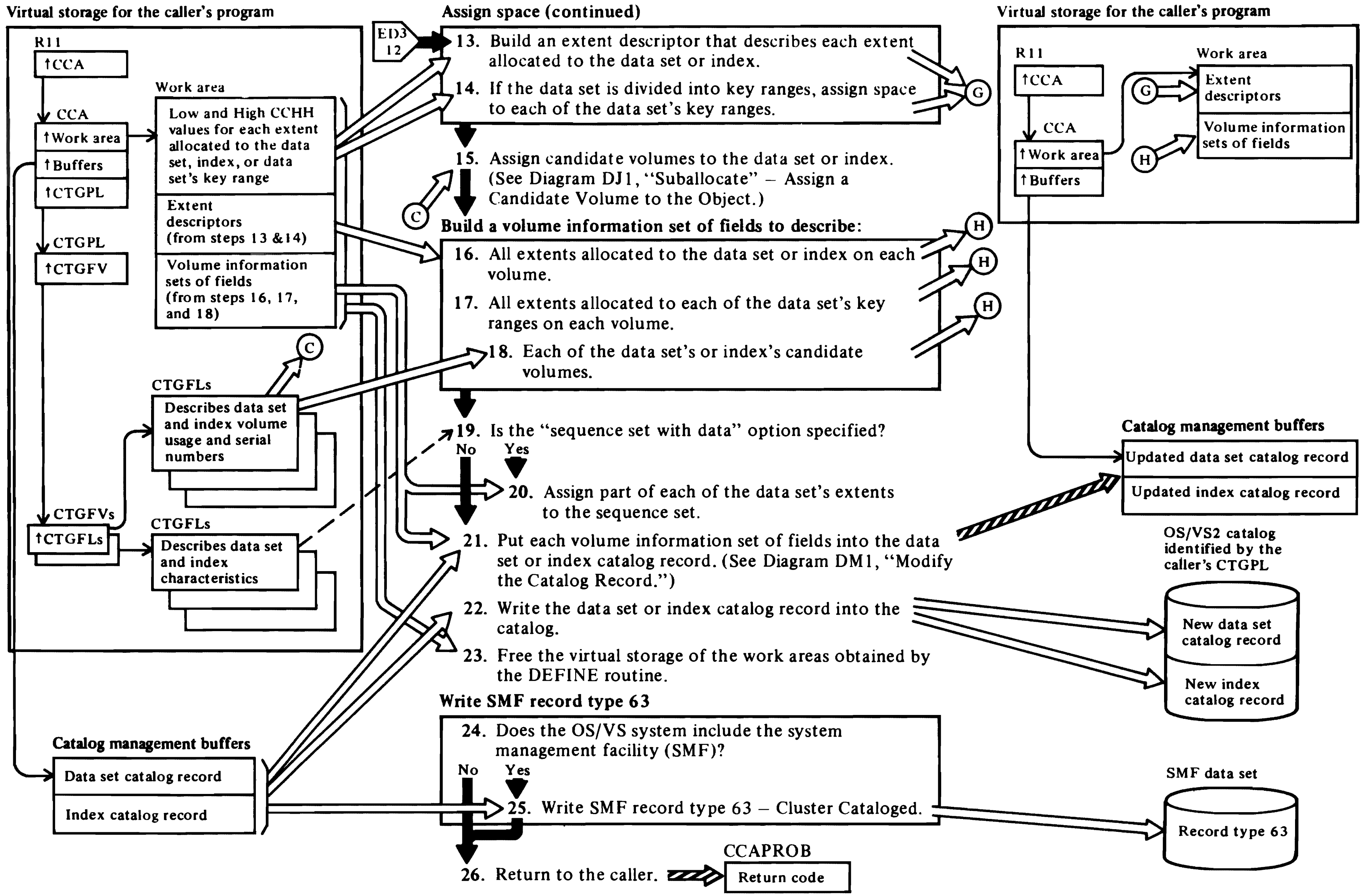
The data set's name is added to the volume catalog record's Data Set Directory Entry set of fields, and the volume catalog record's Data Space Group set of fields is updated to show the cylinders and tracks allocated to the new data set or index.

For additional information about topics related to DEFINE AIX processing, see:

"Data Areas:"

Catalog record description and format
Data set catalog record description and format
Volume catalog record description and format
Data set group set of fields description
Data set directory entry set of fields description
Access Method data set statistics block (AMDSB) description and format

Diagram ED4. DEFINE AIX: Create an Alternate Index



Notes for Diagram ED4

13 IGG0CLAJ: IGGPDSEX

14 IGG0CLAJ: IGGPDSPO

Each key range is assigned physical space and space allocation continues until all key ranges have been assigned space.

15 IGG0CLAJ: IGGPDCNV

A candidate volume is available to contain part of the cluster if more space is needed later. None of the candidate volume's space is allocated to the data set or index when the AIX is created.

16 IGG0CLAK: IGGPDBVO

Each volume that contains a part of the data set or index is described by a volume information set of fields within the data set or index catalog record.

17 IGG0CLAK: IGGPDRNG

If the data set is divided into key ranges, each key range's space on a volume is described in a separate volume information set of fields. If the key range's space is on more than one volume, each volume that contains part of the key range's space is described in a separate volume information set of fields.

18 IGG0CLAK: IGGPDBCV

IGGPDBCV builds a volume information set of fields for each candidate volume of the data set.

20 IGG0CLAK: IGGPDSSP

If the "sequence set with data" option is specified, part of the data set's space is allocated to the index for sequence set records. The low and high CCHH values in the index record's volume information set of fields are those of the extent obtained for the data set. The low and high RBA values are for the sequence set and are relative to the index addresses.

21 IGG0CLAK: IGGPDMOP

IGGPDMOP calls IGGPMOD to add each volume information set of fields to the record.

22 IGG0CLAK: IGGPDMOP

A catalog management routine writes the completed data set or index catalog record into the VSAM catalog and frees the catalog management buffer that contains the record.

23 IGG0CLA8: IGGPDFRS

IGGPDFRS frees all unneeded storage resources.

25 IGG0CLAJ: IGGPDBDI (calls **IGGPSMFA (IGG0CLBV)**)

See *OS/VS2 System Programming Library: System Management Facilities* for details of SMF record type 63—VSAM Data Set Cataloged. Record type 63 is written after a VSAM AIX is defined and whenever the definition is altered.

For additional information about topics related to DEFINE AIX processing, see:

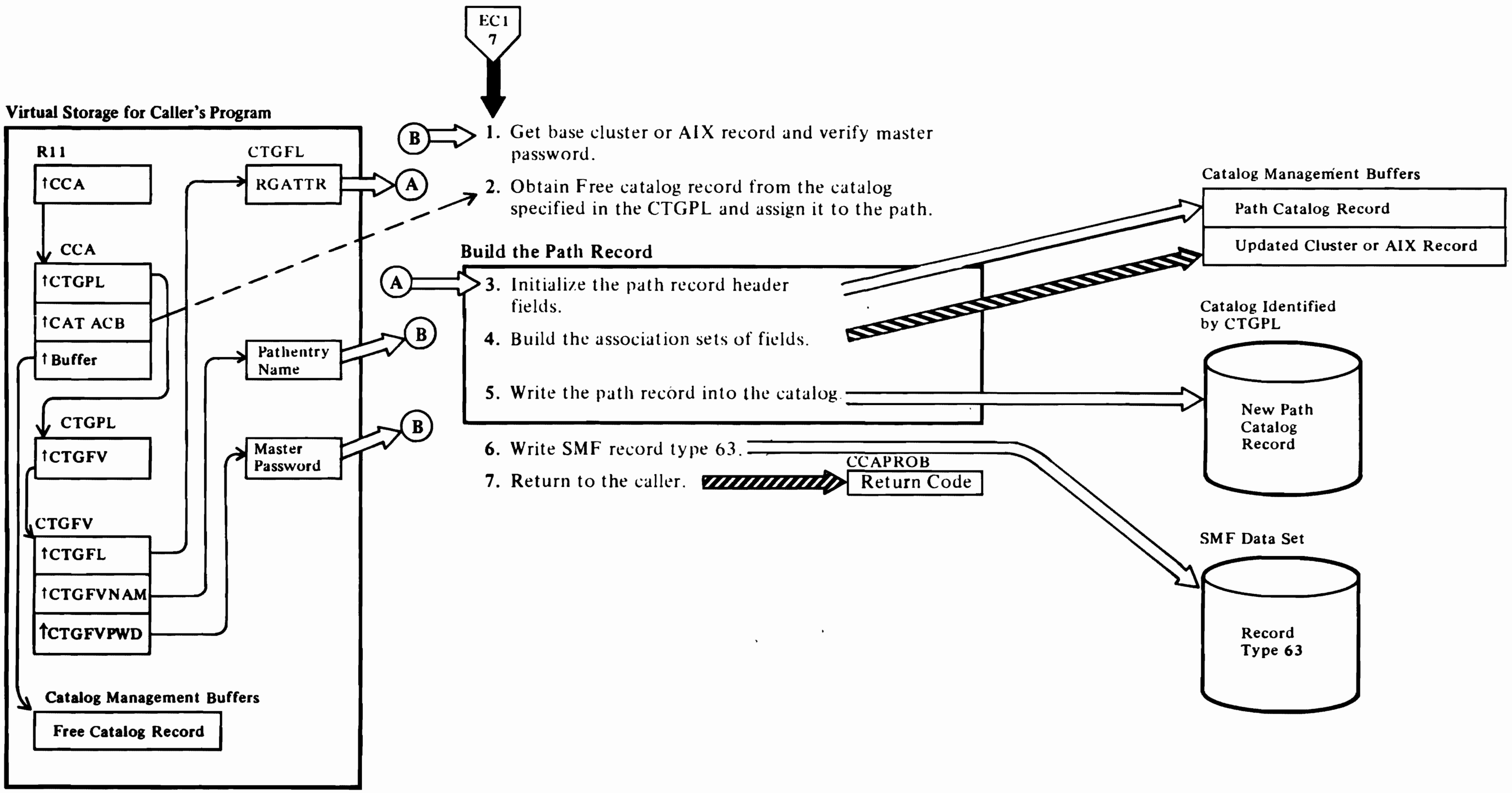
"Data Areas:"

Data set catalog record description and format
Index catalog record description and format
Volume information set of fields description

"Diagnostic Aids:"

Catalog management return codes

Diagram ED5. DEFINE PATH: Create a Path



Notes for Diagram ED5

This figure describes the processes performed by catalog management services routines when the user issues the Access Method Services DEFINE PATH command in the form:

DEFINE PATH (NAME(*name*)
PATHENTRY(*entryname/password*)
(*parameterlist*)

where:

- *name* specifies the name to be given to the alternate index/base cluster pair (i.e., the path).
- *entryname* specifies the name of the alternate index or cluster which is to be considered as the entry to the path.
- *password* specifies the master-level password.
- *parameterlist* is a list of optional and required parameters that define the path's characteristics.

See *OS/VS2 Access Method Services* for details about the DEFINE command.

1 IGG0CLB9: IGGPPRPW

2 IGG0CLAG: IGGPAOCI

IGGPAOCI is called to obtain one catalog control interval to contain the path record.

3 IGG0CLB9: IGGPBAWP

4 IGG0CLB9: IGGPBAWP and **IGGPBAMC**

Builds an association set of fields to point to the cluster and index records and, if the cluster is key-sequenced, the index record. If the path is being built over the AIX, then the pointers to the AIX, the AIX data, and the AIX index records are also included. An association set of fields is also built in the cluster or AIX record to point to the path record.

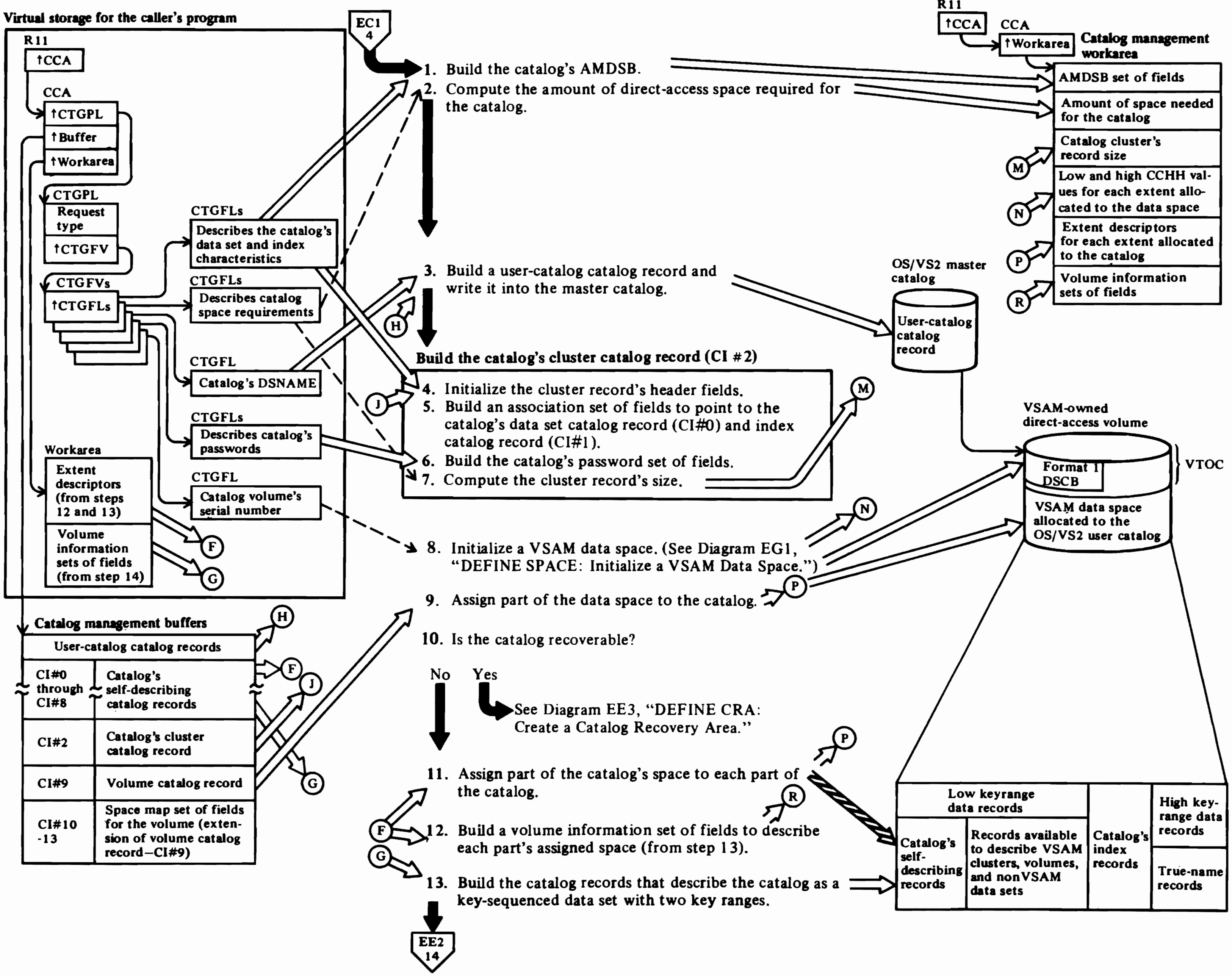
5 IGG0CLB9: IGGPBAWP

The DEFINE routine issues a modify with a put/add option to insert all the association sets of fields into the path record and to write the record into the catalog. These association sets of fields had been extracted previously from the cluster and the AIX records, if the path is being built over an AIX.

6 IGG0CLBV: IGGPSMFA

Record type 63 is written after a VSAM path is defined.

Diagram EE1. DEFINE CATALOG: Create an OS/VS2 Catalog



Notes for Diagram EE1

This figure describes the processes performed by the catalog management services routines when the user issues the Access Method Services DEFINE USERCATALOG command in the form:

DEFINE
USERCATALOG
(PARAMETERLIST)
[CATALOG (catname/password)]

where:

- *catname* is the DSNAME of the OS/VS2 master catalog, which will contain the catalog record that describes the OS/VS2 user catalog.
- *password* is the master catalog's master, control interval, or update password, if the master catalog is protected by passwords.
- *parameterlist* fields are described in *OS/VS2 Access Method Services*.

1 IGG0CLAP: IGGPDCDA

The AMDSB contains the catalog's statistics and fixed characteristics. Each time the catalog is opened, the AMDSB is retrieved from the catalog's data set catalog record (control interval number 0). When the catalog is closed, the AMDSB is usually updated and rewritten into the data set catalog record. VSAM Close processing contains a performance enhancement that determines whether or not the catalog's AMDSB needs to be updated.

2 IGG0CLAP: IGGPDCSP

The field vector table (CTGFV) contains addresses of buffer-size and record-length CTGFLs. This data is used to determine the catalog's control-interval and control-area size, and the amount of space required for the catalog.

3 IGG0CLAP: IGGPDCPC

5 IGG0CLAN: IGGPDCCE

The cluster catalog record contains an associated-data-set set of fields to locate the catalog's data set catalog record (control interval number 0) and an associated-index set of fields to locate the catalog's index catalog record (control interval number 1).

8 IGG0CLAS: IGGPDCSP (calls **IGGPDEFS (IGG0CLAQ)**)

The OS/VS2 catalog is always built in a data space that can contain other VSAM data sets and indexes. A new data space is allocated to VSAM by the OS/VS DADSM Allocate routine, and the data space is assigned to the new catalog.

9 IGG0CLAS: IGGPDCSP(calls **IGGPSALL (IGG0CLAR))**

A Data Set Directory Entry set of fields containing the cluster's control interval number is added to the volume catalog record.

10 IGG0CLAS: IGGPDCSP (calls **IGGPDCRA (IGG0CLB4))**

11 IGG0CLAS: IGGPDCLD

The catalog might contain records that describe user's VSAM data sets, user's nonVSAM data sets, direct-access volumes, and (in the master catalog) user catalogs.

12 IGG0CLAS: IGGPDCVO

13 IGG0CLAS: IGGPDCBE

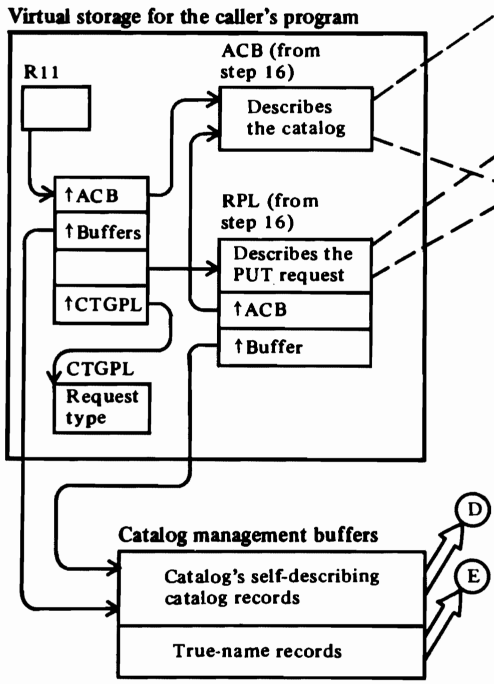
These records define the catalog as a key-sequenced VSAM data set, describe the space allocated to the catalog's data records, index records, and true-name records, describe the free space control intervals within the catalog's data space, and describe the allocated and unallocated tracks on a catalog's volume.

For additional information about topics related to DEFINE CATALOG processing, see:

"Data Areas:"

Catalog record description and format
Cluster catalog record description and format
User-catalog catalog record description and format
Volume catalog record description and format
Data space group set of fields description
Data set directory entry set of fields description
First preformatted records in the catalog
Access method data set statistics block (AMDSB) description and format

Diagram EE2. DEFINE CATALOG: Create an OS/VS2 Catalog



EE1
13

Open the Catalog

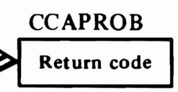
14. Build the control blocks to allow the catalog to be opened and processed as a VSAM data set.
15. Open the catalog's ACB to allow catalog records to be written. (See "Open: Connect a User to a VSAM Data set," in *OS/VS2 VSAM Logic.*)
16. Write the catalog's self-describing records (built in step 15) into the catalog.
17. Build and write true-name catalog records for the catalog DSNAME and the data space's volume serial number.
18. Close the catalog's ACB. (See "Close: Disconnect a User from a VSAM Data Set," in *OS/VS2 VSAM Logic.*)
19. Delete the control blocks built in step 16.

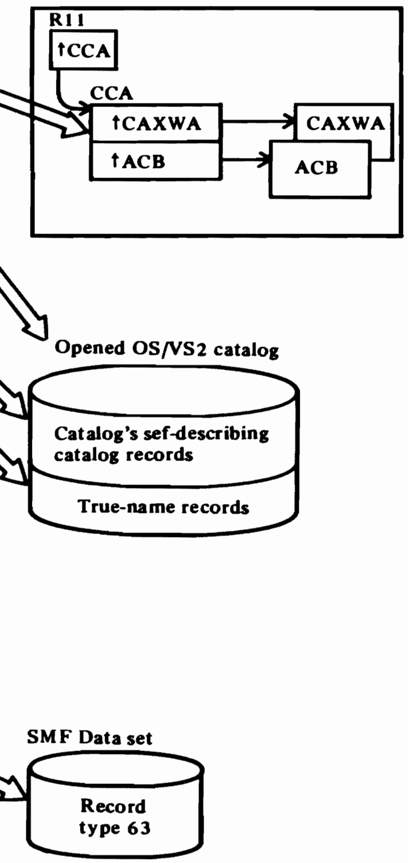
Write SMF record type 63

20. Does the OS/VS system include the system management facilities (SMF)?

No Yes

21. Write SMF record type 63 – Cluster Cataloged.
22. Return to the caller.





Notes for Diagram EE2

14 IGG0CLAE: IGGPDCCB

15 IGG0CLAE: IGGPDCCB

The ACB describes the catalog as a VSAM data set to VSAM record management routines.

16 IGG0CLAE: IGGPDCPR

17 IGG0CLAE: IGGPDCPR

21 IGG0CLBV: IGGPSMFA

See *OS/VS2 System Programming Library: System Management Facilities* for the format of SMF record type 63—VSAM Data Set Cataloged. Record type 63 is written after an OS/VS2 catalog is defined (for the cluster, data, and index components) and whenever the catalog's definition is altered.

22

The catalog management services Define routine sets a return code CCAPROB and returns to the caller whenever an error is detected.

For additional information about topics related to DEFINE CATALOG processing, see:

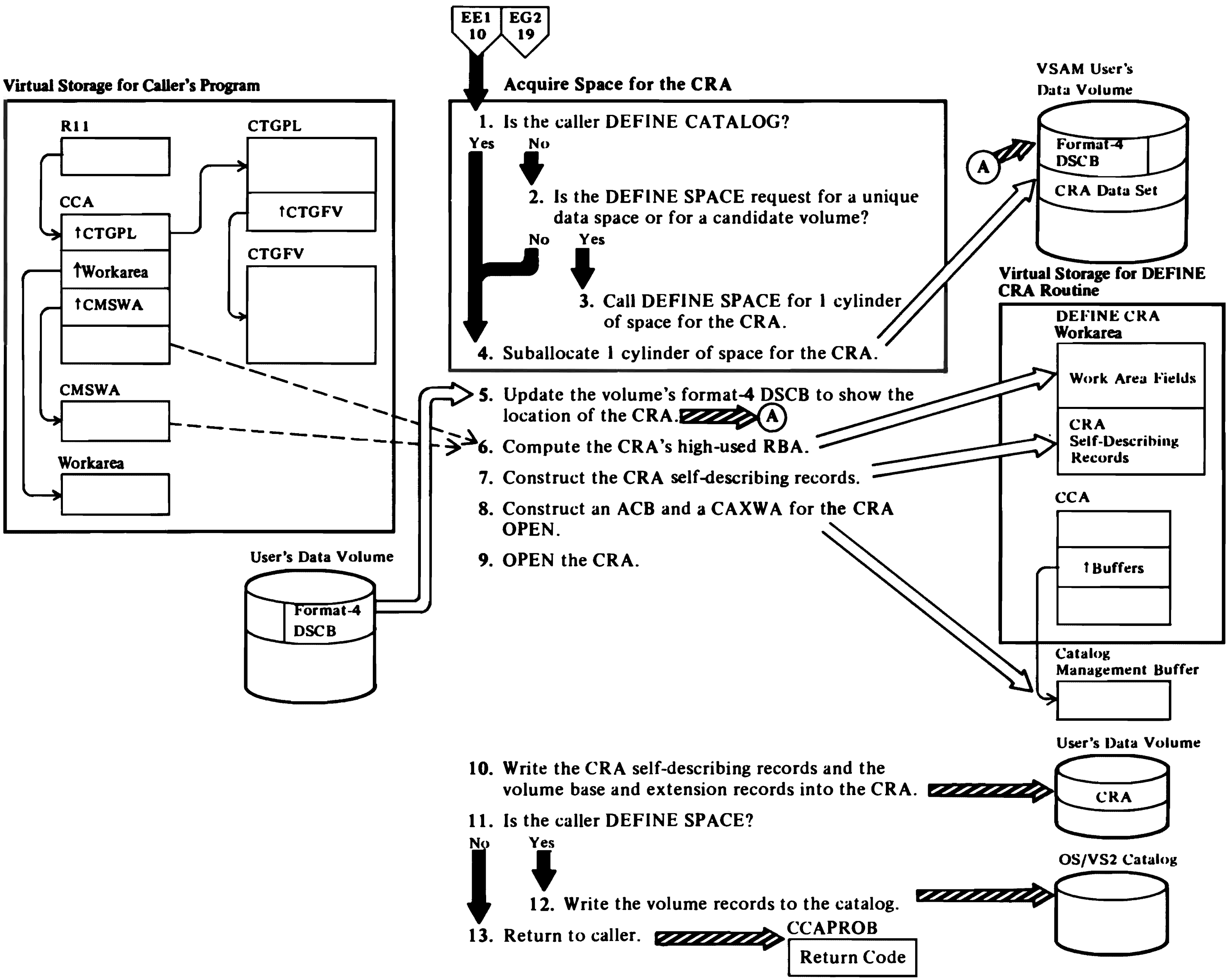
"Data Areas:"

True-name catalog record description and format
Access method control block (ACB) description and format
Catalog auxiliary work area (CAXWA) description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram EE3. DEFINE CRA: Create a Catalog Recovery Area



Acquire Space for the CRA

1. Is the caller DEFINE CATALOG?
   Yes / No
2. Is the DEFINE SPACE request for a unique data space or for a candidate volume?
   No / Yes
3. Call DEFINE SPACE for 1 cylinder of space for the CRA.
4. Suballocate 1 cylinder of space for the CRA.
5. Update the volume's format-4 DSCB to show the location of the CRA. (A)
6. Compute the CRA's high-used RBA.
7. Construct the CRA self-describing records.
8. Construct an ACB and a CAXWA for the CRA OPEN.
9. OPEN the CRA.
10. Write the CRA self-describing records and the volume base and extension records into the CRA.
11. Is the caller DEFINE SPACE?
    No / Yes
12. Write the volume records to the catalog.
13. Return to caller.

Notes for Diagram EE3

1 IGG0CLB4: IGGPSTRG

2 IGG0CLB4: IGGPSTRG

3 IGG0CLB4: IGGPRDEF

If the caller is DEFINE SPACE and the request is for a unique data space or for a candidate volume, IGGPRDEF will construct an interface to recursively call DEFINE SPACE (IGGPDEFS(IGG0CLAQ)) to obtain one cylinder of space for the CRA.

4 IGG0CLB4: IGGPSBAL

Suballocate one cylinder for the CRA from the data space.

5 IGG0CLB4: IGGPFMT4

IGG0CLBU: IGGPF4RD, IGGPF4WR

6 IGG0CLB4: IGGPCHIU

7 IGG0CLB4: IGGPCCIO, IGGPCI15

IGG0CLDA: IGGPMODI

The CRA self-describing records are constructed in core storage; the MODIFY function is called via IGGPMODI to complete the record construction.

8 IGG0CLB4: IGGPCACB, IGGPCXWA

An ACB and a CAXWA are built in a catalog management buffer for use by OPEN.

9 IGG0CLB4: IGGPDCRA

10 IGG0CLDA: IGGPWCRA

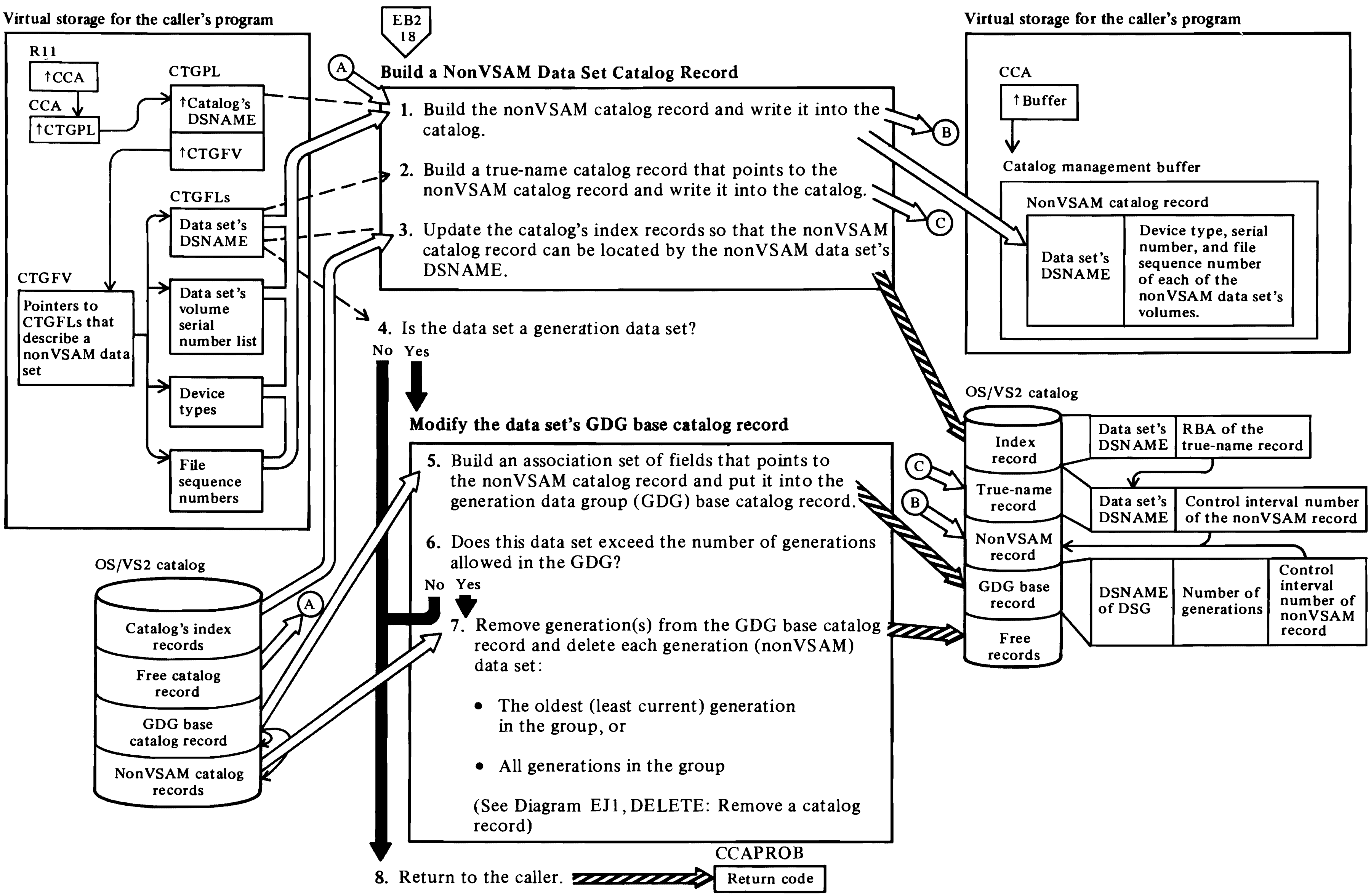
The CRA self-describing records are written to the volume. The volume base and extension records are also written to the CRA. For a DEFINE CATALOG caller, the records are pointed to by an address array in the CMS work area; for the DEFINE SPACE caller, the volume records are chained from a pointer in the CCA.

12 IGG0CLDA: IGGPWCAT

The chained volume records are also written into the catalog if DEFINE SPACE is the caller.

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram EF1. DEFINE NONVSAM: Define a NonVSAM Data Set in an OS/VS2 Catalog



Notes for Diagram EF1

This figure describes the processes performed by the catalog management services routines when the user issues the Access Method Services DEFINE NONVSAM command in the form:

DEFINE
NONVSAM (parameterlist)
[CATALOG (catname/password)]

where:

- *catname* is the DSNAME of the catalog that will contain the nonVSAM catalog record that describes the user's nonVSAM data set.
- *password* is the catalog's master or update password, if the catalog is protected by passwords.
- *parameterlist* fields are described in the appropriate Access Method Services books.

1 IGG0CLBH: IGGPDEFA, IGGPDAIN and **IGGPDAVO**

The DEFINE NONVSAM routine builds and transfers the nonVSAM catalog record from a catalog management buffer in virtual storage to the catalog specified by the caller's DEFINE command.

If the nonVSAM data set is being defined in a recoverable catalog, the catalog's volume serial and device type are saved in the nonVSAM catalog record.

2 IGG0CLBH: IGGPDEFA

3 IGG0CLBH: IGGPDEFA

5 IGG0CLBH

When a data set generation is added to the GDG base catalog record, IGG0CLBH builds an association set of fields that points to the generation data set's catalog record and adds the association set of fields to the GDG base catalog record.

7 IGG0CLBH

If the number of generations (association sets of fields) in the GDG base catalog record exceeds a limit, IGG0CLBH calls IGG0CLBG to delete the earliest generation data set. The user can also specify the deletion of all generations (except the one being defined) in the group when the number of generation data sets exceeds a maximum number.

Note: This figure also describes the processes performed when the user issues an IMPORT command to connect a user catalog (created on another OS/VS system and defined in that system's master catalog) to the master catalog. This process is similar to defining a nonVSAM data set, and should not be confused with the process described in Diagram EE1, DEFINE CATALOG: Create an OS/VS2 Catalog.

The user catalog record is similar to the nonVSAM catalog record, except the record's ID (identifier) value is "U" and there is one volume information set of fields to describe the volume containing the user catalog.

For additional information about topics related to DEFINE NONVSAM processing, see:

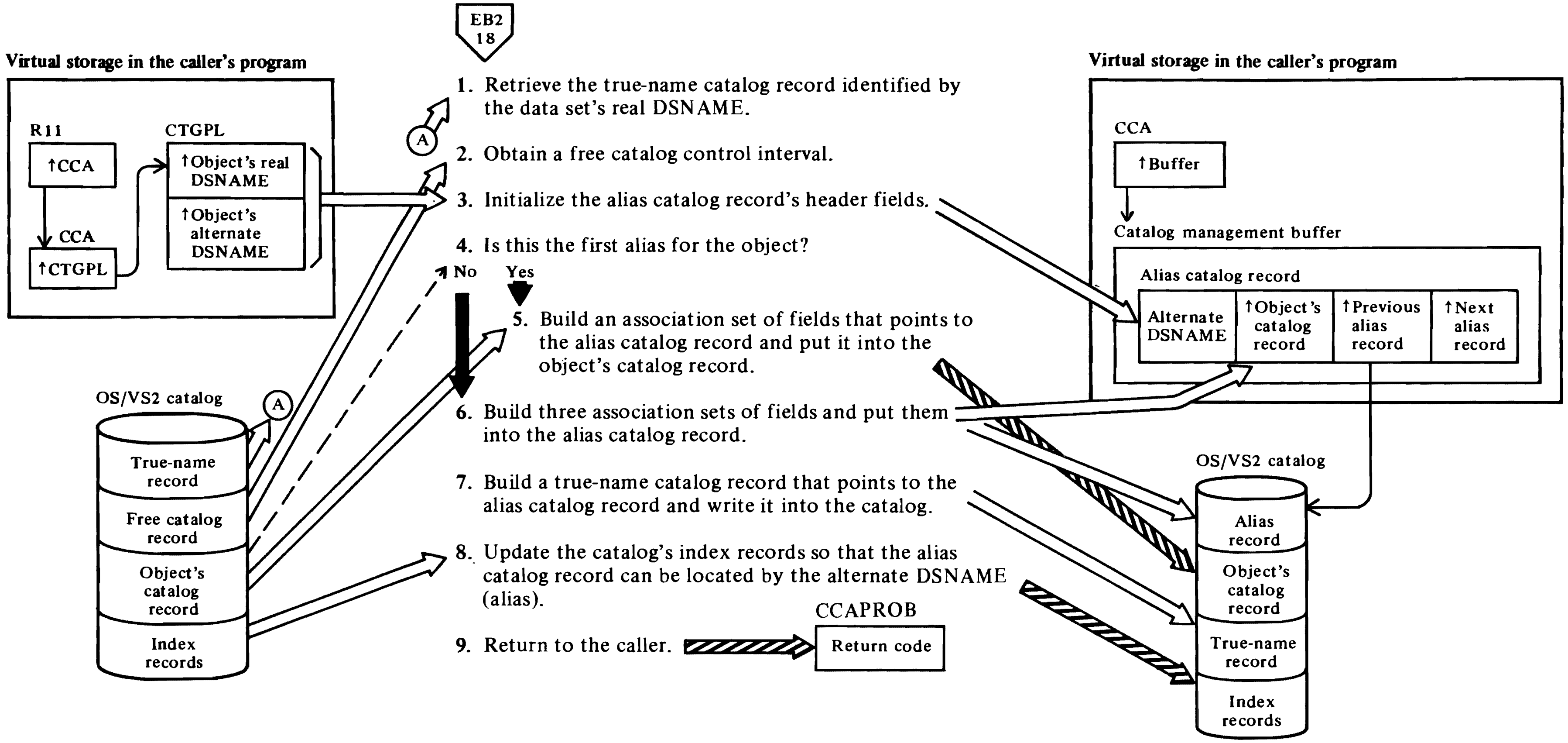
"Data Areas:"

Catalog index record description
NonVSAM catalog record format and description
True-name catalog record description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram EF2. DEFINE ALIAS: Identify a User's Catalog or NonVSAM Data Set with an Alternate DSNAME



1. Retrieve the true-name catalog record identified by the data set's real DSNAME.
2. Obtain a free catalog control interval.
3. Initialize the alias catalog record's header fields.
4. Is this the first alias for the object?
5. Build an association set of fields that points to the alias catalog record and put it into the object's catalog record.
6. Build three association sets of fields and put them into the alias catalog record.
7. Build a true-name catalog record that points to the alias catalog record and write it into the catalog.
8. Update the catalog's index records so that the alias catalog record can be located by the alternate DSNAME (alias).
9. Return to the caller.

Notes for Diagram EF2

This diagram describes the processes performed by catalog management services routines when the user issues the Access Method Services DEFINE ALIAS command in the form:

DEFINE
ALIAS
(parameterlist)
[CATALOG (catname/password)]

where:

- *catname* is the DSNAME of the catalog that contains the nonVSAM data set's catalog record for which this command will specify an alias name.
- *password* is the catalog's master, control interval, or update password, if the catalog is protected by passwords.
- *parameterlist* fields are described in the appropriate Access Method Services books.

The DEFINE ALIAS command is used to establish an alternate DSNAME for a nonVSAM data set or user's catalog. The command cannot be used with VSAM data sets, data spaces, or the master catalog.

If the alias is being defined in a recoverable catalog, the catalog's volume serial and device type are saved in the alias catalog record.

1 IGG0CLAO: IGGPDEFX

The data set's or user catalog's real dsname is used to locate its true name catalog record.

2 IGG0CLAO: IGGPDEFX (calls **IGG0CLAG**): **(IGGPAOCI)**

3 IGG0CLAO: IGGPDEFX

5 IGG0CLAO: IGGPBCFI (calls **IGG0CLAV**): **(IGGPMOD)**

6 IGG0CLAO: IGGPBCFI (calls **IGG0CLAV**): **(IGGPMOD)**

IGGPBCFI builds association sets of fields to point to:

- The object's catalog record (containing the object's real DSNAME).
- The previously-built alias name catalog record for the object. If this is the object's first alias, this set of fields contains a control interval number (pointer) equal to zero.
- The next alias name catalog record for the object. This set of fields contains a control interval number (pointer) equal to zero. When the next alias for the object is specified (with a DEFINE ALIAS command), this set of fields is updated to point to the newly-built alias name catalog record.

7 IGG0CLAO: IGGPDEFX

For additional information about topics related to DEFINE ALIAS processing, see:

"Data Areas:"

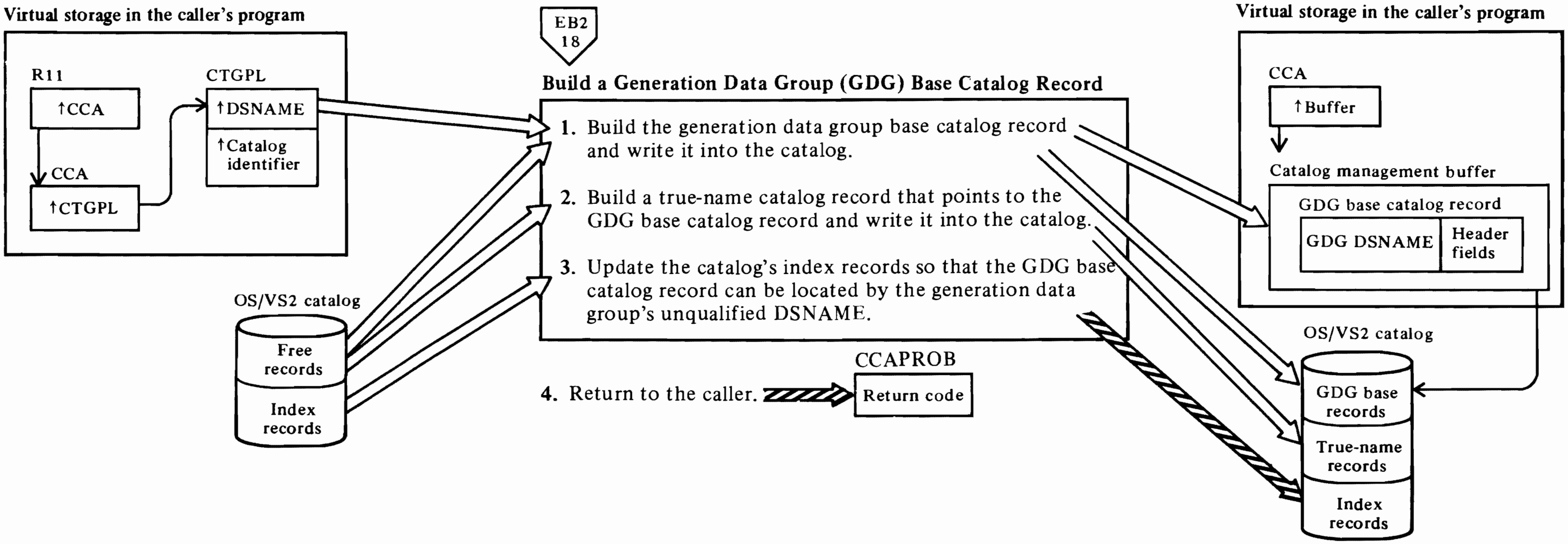
Catalog parameter list (CTGPL) description and format
Alias name catalog record description and format
True-name catalog record description and format

"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram EF3. DEFINE GENERATION DATAGROUP: Create a Generation Data Group



Notes for Diagram EF3

This diagram describes the processes performed by the catalog management services routines when the user issues the Access Method Services DEFINE GENERATIONDATAGROUP in the form:

DEFINE
GENERATIONDATAGROUP
(parameterlist)
[CATALOG (catname/password)]

where:

- *catname* is the name of the catalog that is to contain the GDG Base catalog record which this command creates.
- *password* is the catalog's master, control interval, or update password, if the catalog is protected by passwords.
- *parameterlist* fields are described in *OS/VS2 Access Method Services*.

The DEFINE GENERATIONDATAGROUP command is used to create a generation data group (GDG). A generation data group is a collection of nonVSAM data sets that have similar DSNAMEs, in the form 'dsname.GxxxxVyy'. All data sets in the group have the same 'dsname'. Each data set is uniquely identified by its generation number 'Gxxxx' and its version number 'Vyy'. Each data set (a generation data set) is described by a nonVSAM catalog record identified by its absolute DSNAME ('dsname.GxxxxVyy'). The user's program can refer to a generation data set with its relative DSNAME, 'dsname(n)', where 'n' is a negative or positive integer.

The DEFINE GENERATIONDATAGROUP command builds a generation data group (GDG) base catalog record. The GDG Base catalog record is used to identify all generation data sets described in the catalog.

1 IGG0CLA4: IGGPDEFB

The GDG Base catalog record header fields are initialized with the generation data group attributes, the maximum number of generation data sets allowed to exist at the same time, and the size of the variable-length generation-level table. The generation data group's DSNAME identifies the GDG base catalog record.

If the GDG base record is being defined in a recoverable catalog, the catalog's volume serial and device type are saved in the GDG base catalog record.

2 IGG0CLA4: IGGPDEFB

When IGGPDEFB finishes completes the initialization of the GDG Base record, IGGPDEFB issues the ADDREC macro instruction to write the GDG Base record into the catalog, to build and write a true name catalog record, and to update the catalog's index records. The ADDREC macro instruction results in a call to IGGPPAD (IGG0CLAG).

For additional information about topics related to DEFINE GENERATIONDATAGROUP processing, see:

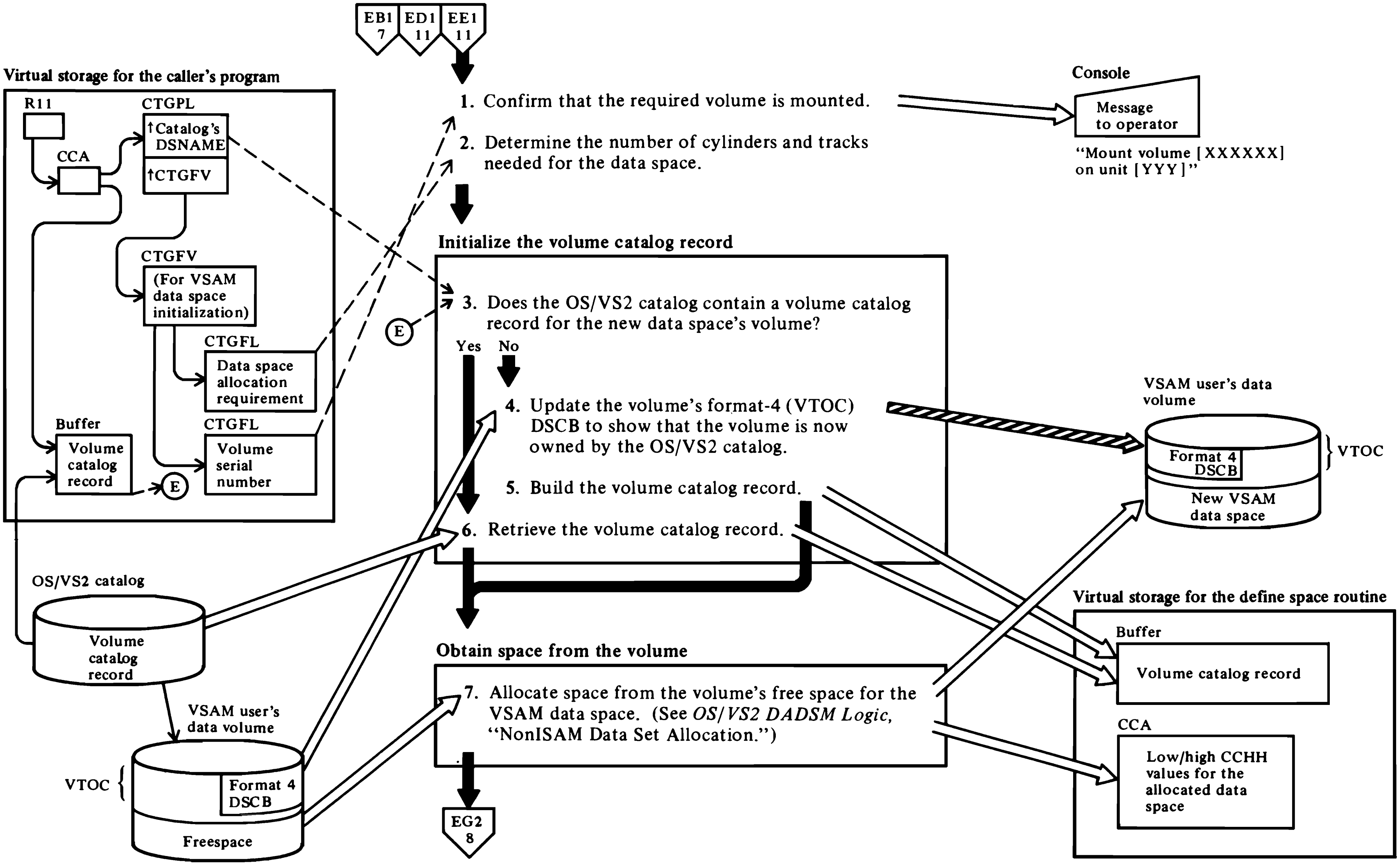
"Data Areas:"

GDG Base catalog record
True Name catalog record
Catalog parameter list (CTGPL) description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram EG1. DEFINE SPACE: Initialize a VSAM Data Space



1. Confirm that the required volume is mounted.
2. Determine the number of cylinders and tracks needed for the data space.

Initialize the volume catalog record

3. Does the OS/VS2 catalog contain a volume catalog record for the new data space's volume?
4. Update the volume's format-4 (VTOC) DSCB to show that the volume is now owned by the OS/VS2 catalog.
5. Build the volume catalog record.
6. Retrieve the volume catalog record.

Obtain space from the volume

7. Allocate space from the volume's free space for the VSAM data space. (See *OS/VS2 DADSM Logic*, "NonISAM Data Set Allocation.")

Notes for Diagram EG1

This diagram describes the processes performed by the catalog management services routines when the user issues the Access Method Services DEFINE SPACE command:

DEFINE
SPACE (parameterlist)
[CATALOG (catname/password)]

where:

- *catname* is the DSNAME of the catalog that contains the volume catalog record that will describe the VSAM data space.
- *password* is the catalog's master, control interval, or update password, if the catalog is protected by passwords.
- *parameterlist* fields are described in *OS/VS2 Access Method Services*.

1 IGG0CLAQ: IGGPDEFS and **IGGPVMTV**

2 IGG0CLA6: IGGPCRTC

The user can specify the data space's cylinder and track requirements, or he can specify a number of records and the length of each record, to define the data space's allocation requirements.

3 IGG0CLAQ

If a volume catalog record exists for the volume, and if the volume already contains a VSAM data space, a Data Space Group set of fields is added to the volume catalog record to describe the new VSAM data space. A new format-1 (identifier) DSCB is added to the volume's VTOC to describe the new extent.

4 IGG0CLAQ: IGGPCOBT

If the volume is a candidate volume (one that is eligible to contain a VSAM data space, but doesn't yet) the volume's format-4 (VTOC) DSCB is updated to show that the VSAM catalog is now the volume owner.

5 IGG0CLAQ: IGGPIVER

6 IGG0CLAQ: IGGPDEFS (calls **IGGPSCAT (IGG0CLAH)**)

The volume catalog record is identified by the volume's serial number.

7 IGG0CLA6: IGGPBJFB

IGG0CLAQ: IGGPDEFS

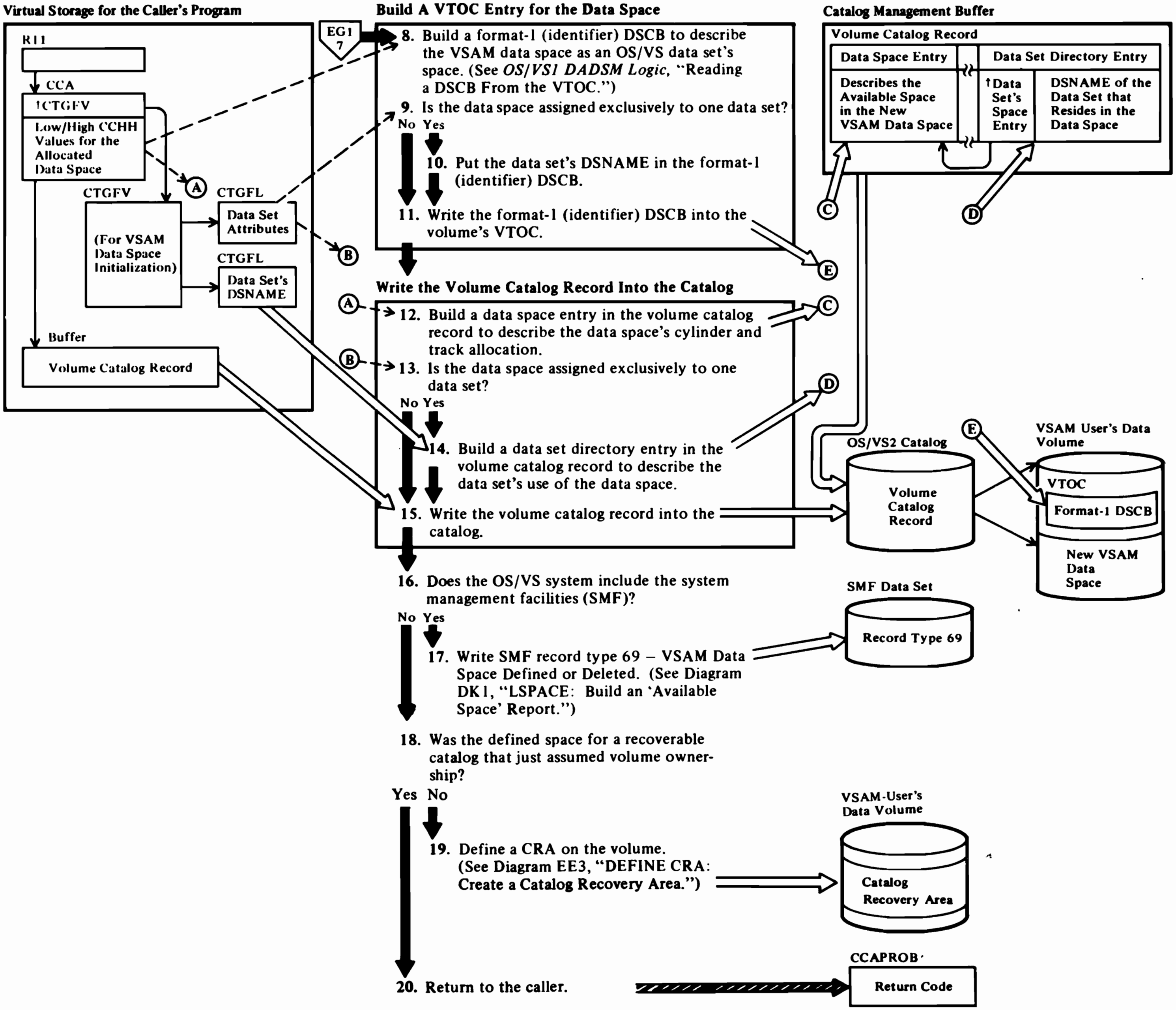
Construct a JFCB and call the DADSM allocate function.

For additional information about topics related to DEFINE SPACE processing, see:

"Data Areas:"

Volume catalog record description and format

Diagram EG2. DEFINE SPACE: Initialize a VSAM Data Space



Notes for Diagram EG2

8 IGG0CLAQ: IGGPDEFS

The format-1 (identifier) DSCB describes the VSAM data space as an OS/VS data set to the OS/VS DADSM routines.

9 IGG0CLAQ: IGGPCOBT

10 IGG0CLAQ: IGGPCOBT

A VSAM data space assigned exclusively to one data set is, to DADSM, the same as one of the extents of an OS/VS data set. The data space is described by a format-1 (identifier) DSCB that contains the data set's dsname. If a data space can be assigned to more than one data set, its format-1 (identifier) DSCB contains a dsname generated by the DEFINE SPACE routine.

11 IGG0CLAQ: IGGPCOBT

12 IGG0CLA6: IGGPCSHG and **IGGPCSDG**

14 IGG0CLA6: IGGPCDSD

The volume catalog record contains the identifier of each data set that resides (in part or in full) on the volume.

17 IGG0CLAQ: IGGPDEFS (calls **IGGPLSP (IGG0CLBK)**)

See *OS/VS2 System Programming Library: System Management Facilities* for the format of SMF record type 69—VSAM Data Space Defined or Deleted. Record type 69 is written when a VSAM data space is created or when its available space is allocated to a VSAM data set or index.

19 IGG0CLB4: IGGPDCRA

A catalog recovery area is defined on the volume on which the space is being defined if ownership is taken by a recoverable catalog and the defined space does not contain the catalog.

For additional information about topics related to DEFINE SPACE processing, see:

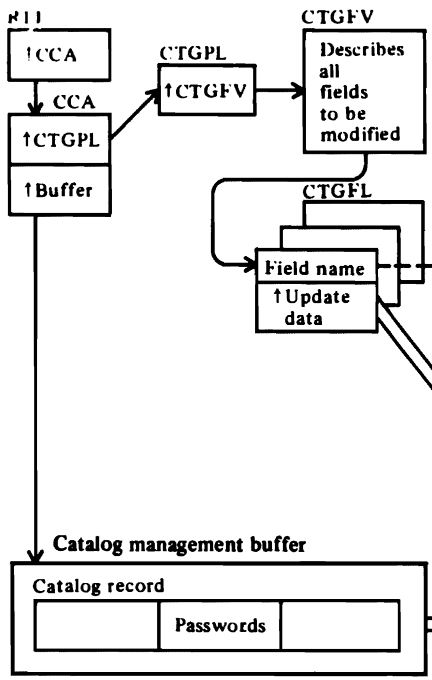
"Data Areas:"

Volume catalog record description and format
Data set group set of fields description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram EH1. ALTER: Modify a Catalog Record

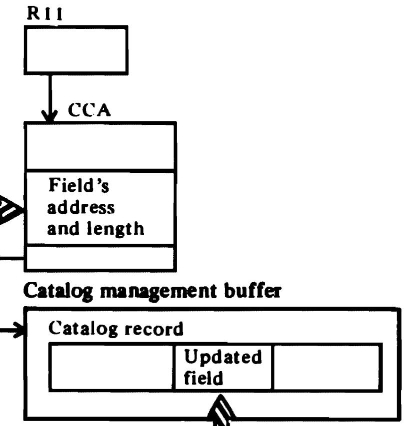


EB1
21

Modify the Catalog Record's Fields

Do steps 1 through 3 for each user-specified field:

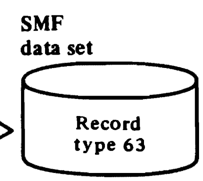
1. Locate the field's address and length. (See Diagram DL1, "Obtain a Catalog Record Field's Value.")
2. Do specialized processing:
   - Change a data set's DSNAME — do steps 8 through 14
   - Perform volume cleanup — do steps 15 through 19
   - Assign a candidate volume — do steps 20 through 22
   - Delete a candidate volume — do steps 23 through 24
3. Modify other catalog fields as specified, replacing the existing information with the updating information. (See Diagram DM1, "Modify a Catalog Record Field's Value.")



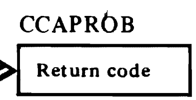
4. Rewrite the updated catalog record.
5. Does the OS/VS system include the system management facilities (SMF)?

No Yes

6. Write SMF record type 63 – VSAM Data Set Cataloged.



7. Return to the caller.



Notes for Diagram EH1

The ALTER command enables the user to modify some of the information he established when he created a VSAM data set.

This figure describes the processes performed by catalog management services routines when the user issues the Access Method Services ALTER command in the form:

ALTER
(entryname/password)
| CATALOG (catname/password) |
other parameters

or:

ALTER
entryname/password
FILE (ddname)
REMOVEVOLUMES (volser[ƀvolser...])

where:

- *entryname* is the DSNAME or volume serial number that identifies the catalog record to be modified.
- *password* is the record's (identified by *entryname*) master password.
- *CATALOG* identifies the catalog that contains the catalog record to be modified, and supplies the correct password for the catalog.
- *other parameters* are described in *OS/VS2 Access Method Services*.

or:

- *entryname,* for the volume cleanup function, is the name of the OS/VS2 master catalog.
- *password* is the master password of the master catalog specified in *entryname.*
- *ddname* specifies the name of a DD statement that identifies the volume(s) to be scratched. This parameter is required for the volume cleanup function.
- *volser* specifies the volume serial number(s) of the volume(s) on which all VSAM space is to be removed and VSAM ownership is to be relinquished. Volumes owned by the master catalog cannot be specified on a cleanup request.

Note: When a user issues the OS catalog management command RECAT, the Catalog Controller converts the user-supplied CAMLST into a nonVSAM volume list. In this case only, all existing volume information sets of fields in the NonVSAM catalog record are deleted and all volumes in the nonVSAM volume list are added to the NonVSAM catalog record as volume information sets of fields.

1 IGG0CLBD: IGGPALT

2 IGG0CLBD: IGGPALT

When the data set name or allocated candidate volumes are changed (in records other than nonVSAM catalog records), other catalog records besides the object's catalog record must be updated.

3 IGG0CLBD: IGGPALT

4 IGG0CLBD: IGGPALNM

6 IGG0CLBD: IGGPALT

See *OS/VS2 System Programming Library: System Management Facilities* for the format of SMF record type 63—VSAM Data Set Cataloged. Record type 63 is written after a nonVSAM data set, cluster, or catalog is defined and when the definition is altered. One SMF record is written for each modified catalog record.

7 IGG0CLBD: IGGPALT

For additional information about topics related to ALTER processing, see:

"Data Areas:"

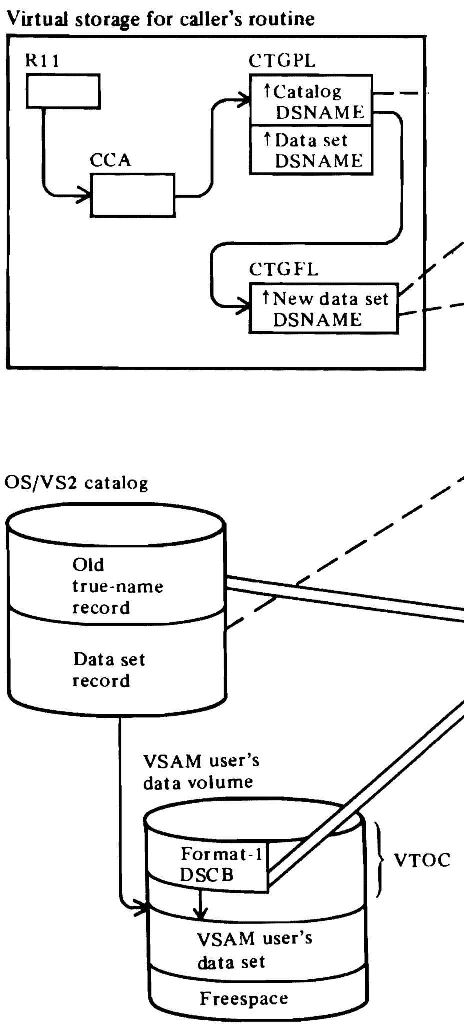
Catalog communications area (CCA) description and format
Catalog parameter list (CTGPL) description and format
Field parameter list (CTGFL) description and format

"Diagnostic Aids:"

Catalog management return codes

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram EH2. ALTER: Modify a Catalog Record



EH1 2

Change a Data Set's DSNAME

8. Search the OS/VS2 catalog to ensure that the new name doesn't duplicate an existing name. (See Diagram DC1, "Search: Retrieve a Catalog Record.")

8.1. If the old name is RACF-indicated, alter (rename) the RACF profile.

9. Build a new true-name catalog record, write it into the OS/VS2 catalog, and update the catalog's index.

10. Does the data set share its data space with other sets?

Yes No

11. Update the data space's format-1 DSCB with the new DSNAME.

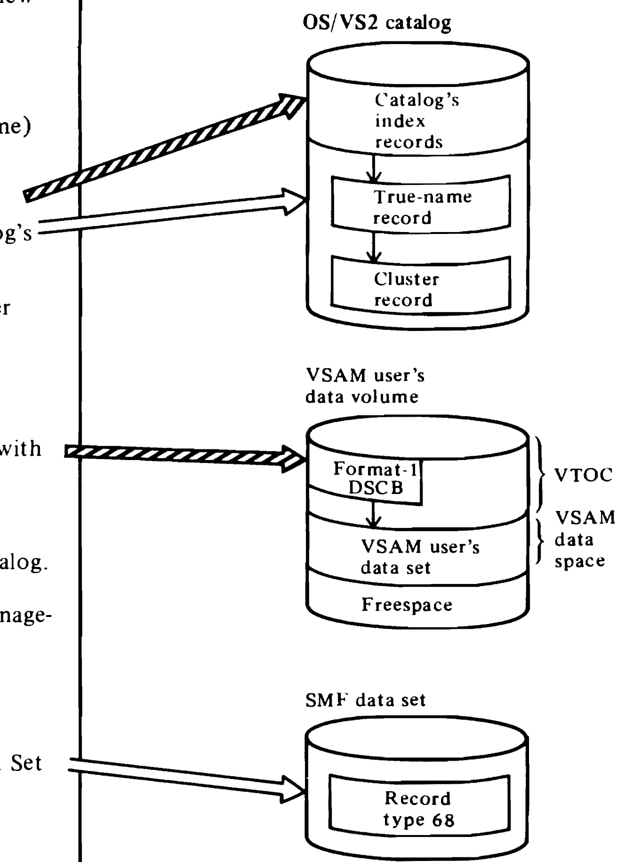
12. Remove the old true-name record from the catalog.

13. Does the OS/VS system include the system management facilities (SMF)?

Yes No

14. Write SMF record type 68 – VSAM Data Set Renamed.

EH1 3



Notes for Diagram EH2

8 IGG0CLBD: IGGPALNM

The catalog specified by the ALTER command's CATALOG parameter is examined. The new DSNAME cannot already exist in the catalog.

8.1 IGG0CLBD:IGGPALNM (calls IGGPALRN (IGG0CLDD))

9 IGG0CLBD: IGGPALNM

A new true name catalog record is put in the high-key range of the catalog.

11

All volumes that contain a format 1 DSCB must have their names modified in the VTOC label.

See *OS/VS2 Data Areas* for details about the DSCB.

12 IGG0CLBD: IGGPALF1

The name and control interval fields in the data set's true-name record are set to 0, and the true-name record's identifier field is set to "F".

14 IGG0CLBD: IGGPALT

See *OS/VS2 System Programming Library: System Management Facilities* for the format of SMF record type 68—VSAM Data Set Renamed.

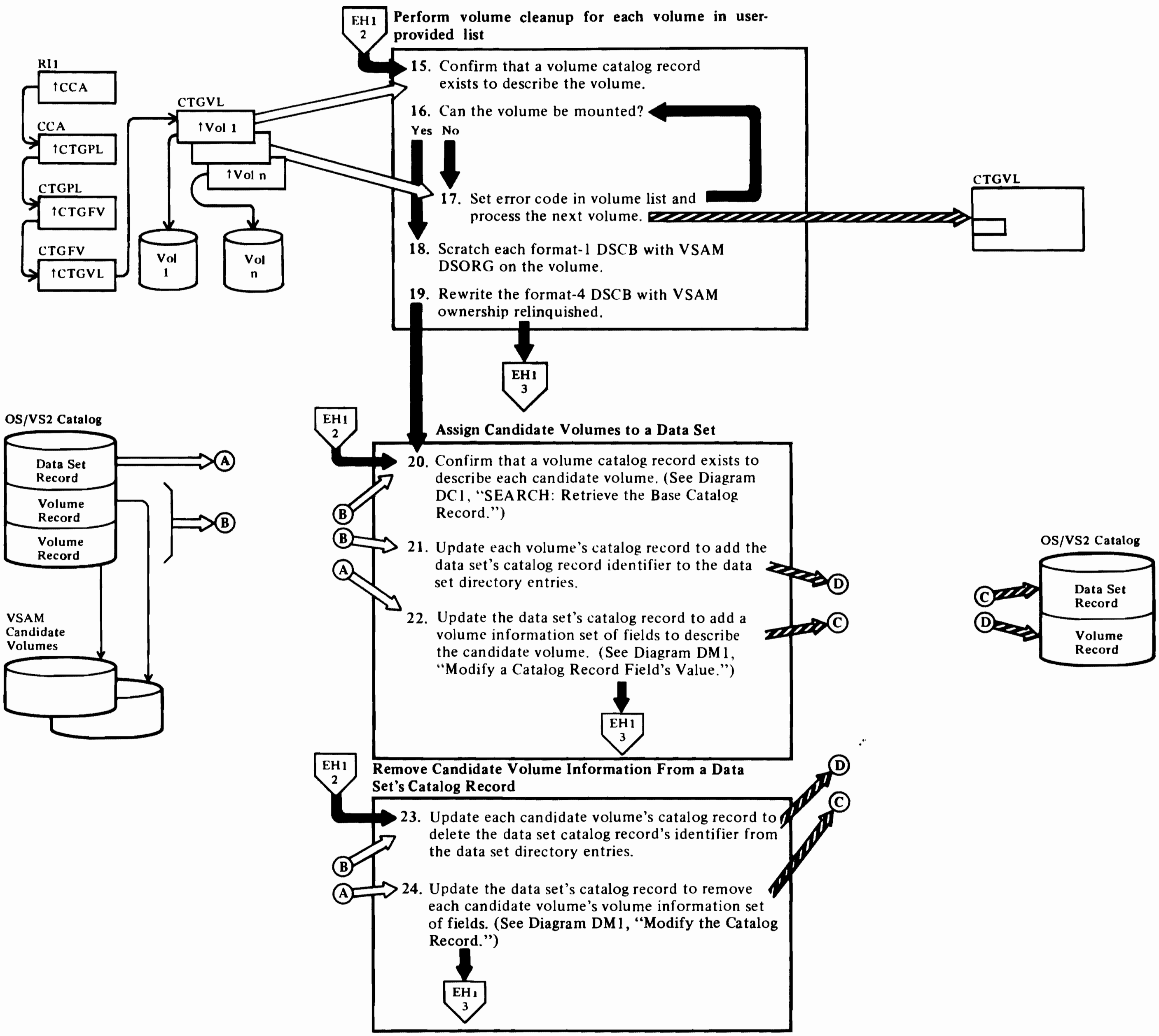
For additional information about topics related to ALTER processing, see:

"Data Areas:"

OS/VS2 catalog organization
Catalog index record description
True-name catalog record description and format
Data set catalog record description and format

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram EH3. ALTER: Modify a Catalog Record



Notes for Diagram EH3

15 IGG0CLBE: IGGPALVL

If volume cleanup is required, IGGPALVL calls IGGPVRD (IGG0CLBN). Each volume in the user-provided volume serial number list is processed, if its volume catalog record exists.

16 IGG0CLBN: IGGPVRD

The volume(s) specified in the user-provided volume serial number list is mounted. After a successful mount, IGGPVRD calls IGGPVRCV (IGG0CLBN).

17 IGG0CLBN: IGGPVRD

If a volume cannot be mounted, a volume-not-mounted condition is indicated in the volume serial list and passed back to the user.

18 IGG0CLBN: IGGPVRCV

Scratch each format-1 DSCB that has VSAM DSORG indicated.

20 IGG0CLBD: IGGPALT (calls **IGGPALVL (IGG0CLBE)**)

IGG0CLBE: IGGPALVA

IGGPALVA calls IGGPSALL (suballocate) to assign the candidate volume to the data set. If a volume catalog record does not exist for the candidate volume, the suballocate routine returns an error code.

See Diagram EG1, DEFINE SPACE: Assign a VSAM Data Space to a Catalog, for details on how a volume catalog record is built.

21 IGG0CLBE: IGGPALVA

The volume catalog record contains a data set directory that describes each VSAM data set's use of the volume's VSAM space.

22 IGG0CLBE: IGGPALVA (calls **IGGPALSA (IGG0CLBE)**, **IGGPSALL (IGG0CLAR)**, and **IGGPMOD (IGG0CLAV)**)

23 IGG0CLBN: IGGPALVR

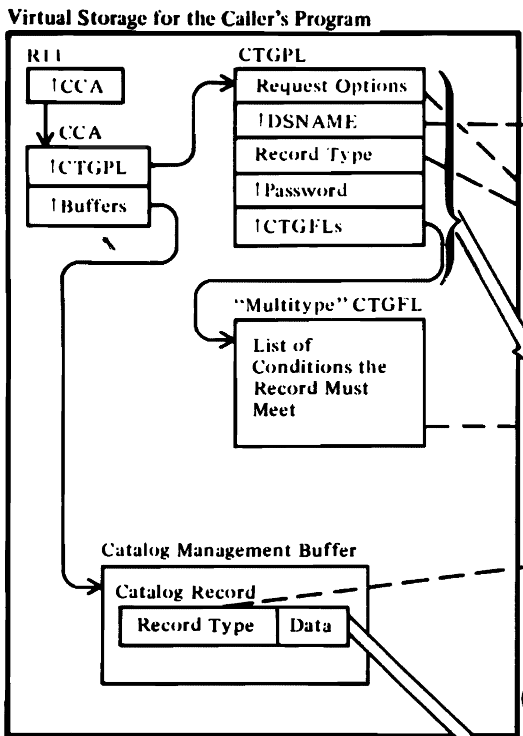
24 IGG0CLBN: IGGPALVR (calls **IGGPMOD (IGG0CLAV)**)

For additional information about topics related to ALTER processing, see:

"Data Areas:"

Data set catalog record description and format
Volume catalog record description and format
Volume information set of fields description and format

Diagram EI1. LISTCAT: Retrieve a Catalog Record's Contents



EB3
21

1. Did the caller supply the catalog's DSNAME?

Yes No

2. Return the DSNAME to the caller. (B)

3. Did the caller specify the CTGPL "GET NEXT" option with record type nonzero?

No. Yes

4. Was "Multitype" CTGFL specified?

No Yes

5. If the catalog record meets the conditions specified in the "multitype" CTGFL, go to step 8; otherwise go to step 7.

6. Is the retrieved catalog record the same type as the CTGPL record type?

Yes No

7. Retrieve the next catalog record and go to step 4. (C)

(A)

8. Verify the user's authorization to read the catalog information requested.

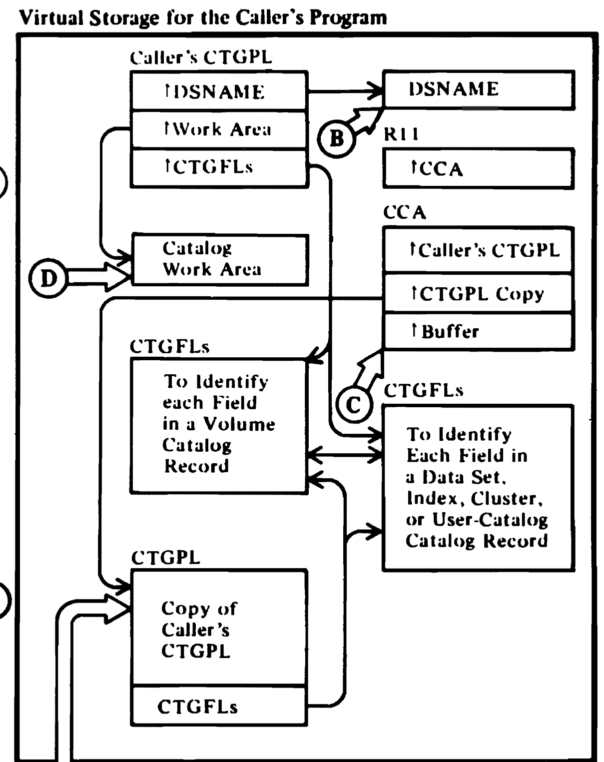
9. Build a copy of the caller's CTGPL and point to the CTGFLs required by this entry.

10. Obtain the catalog record field's values that satisfy the caller's request (see Diagram DL. "Obtain a Catalog Record Field's Value"). (D)

11. Return to caller.

CCAPROB

Return Code



Notes for Diagram EI1

The LISTCAT command enables the user to list all or a part of an OS/VS2 user or master catalog's contents. This figure describes the processes performed by the catalog management services routines when the user issues the Access Method Services LISTCAT command in the form:

LISTCAT
CATALOG (catname[/password][dname])]
[OUTFILE (dname) |
[ENTRIES (entryname[/password]) | LEVEL(level)]
[CLUSTER | DATA | INDEX | SPACE |
NONVSAM | USERCATALOG | PATH |
ALIAS | PAGESPACE | ALTERNATEINDEX |
GENERATIONDATAGROUP]
[ALL | NAME | VOLUME | ALLOCATION |
HISTORY]
[CREATION (days)]
[EXPIRATION (days)]

[NOTUSABLE]

where:

- *CATALOG* identifies the OS/VS2 catalog that contains the user-requested data:
  - *catname* is the DSNAME of a user catalog or the master catalog.
  - *password* is one of the catalog's passwords, if the catalog is password protected. If the user requests password information from the catalog, he must specify the catalog's master password.All other catalog information is available to the user if he specifies the catalog's read password.
- *dname* specifies the DD name of the catalog to be listed.
- *OUTFILE* specifies an optional alternate listing output data set and identifies it by dname.
- *ENTRIES* is a list of catalog record identifiers:
  - *entryname* is the DSNAME or volume serial number that identifies a catalog record. If the LISTCAT command includes an ENTRIES parameter list, only those catalog records identified by entrynames are listed.
  - *password* is one of the catalog record's passwords. If the catalog's password is supplied, the catalog record's password is ignored. Otherwise, the catalog record's master password allows its password information to be listed; its read password allows all other information to be listed, but suppresses the password information.

* LEVEL (level) specifies the level of entrynames to be listed.

- *CLUSTER | DATA | INDEX | SPACE | NONVSAM | USERCATALOG | PATH | ALIAS | PAGESPACE | ALTERNATEINDEX | GENERATIONDATAGROUP* is a list that specifies the types of catalog records to be listed. If both the ENTRIES and this "types" parameter list are specified, only those catalog records that are identified by an entryname and are included in the list of types are returned to the caller.
- *ALL | NAME | VOLUME | ALLOCATION | HISTORY* specifies what part of each record to list.
- *CREATION* specifies the minimum age an object must be to be listed.
- *EXPIRATION* specifies the maximum number of days remaining before expiration of an object may have to be listed.
- NOTUSABLE specifies that only "not usable" data and index entries are to be listed.

The LISTCAT parameters are described in *OS/VS2 Access Method Services*.

1 IGG0CLBQ: IGGPLSTC

If the first character of the catalog's name is blank, the caller wants the catalog name returned.

3 IGG0CLBQ: IGGPLSTC

If the caller did not specify the CTGPL "GET NEXT" option and a nonzero CTGPL record type, only the one original entry pointed to by the CTGPL is listed.

5 IGG0CLBQ: IGGPLSTC

The multitype CTGFL specifies conditions which must be met by the retrieved record. The possible conditions are:

- Must be of a specified record type
- Must have a certain usability state
- Must meet a creation date value
- Must meet an expiration date value

6 IGG0CLBQ: IGGPLSTC

If there is no multitype CTGFL, the retrieved record must be the same record type as the CTGPL record type.

7 IGG0CLBQ: IGGPLSTC

The next record is retrieved by specifying the GET NEXT option to VSAM Record Management.

9 IGG0CLBQ: IGGPLSTC

If the entry is a volume record, only volume CTGFLs are pointed to from the CTGPL; otherwise, only nonvolume CTGFLs are pointed to from the CTGPL.

11 IGG0CLBQ: IGGPLSTC

When all requested information has been retrieved, the LISTCAT routine sets a return code in CCAPROB and returns to the caller, Catalog Management Services Common Processing.

For additional information about topics related to LISTCAT processing, see:

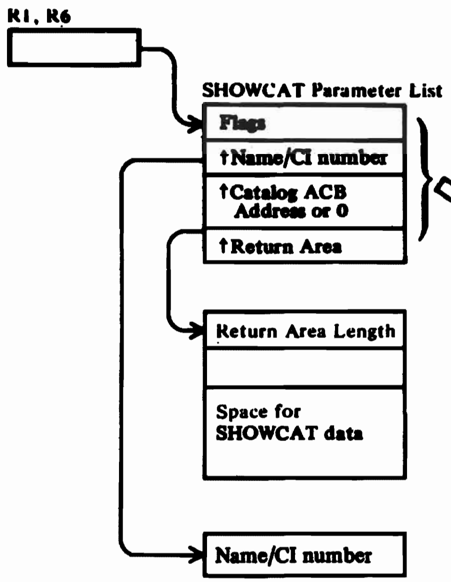
"Data Areas:"

Field parameter list (CTGFL) format and description
Catalog record format and description

"Diagnostic Aids:"

Catalog management return codes

Diagram EI2. Show Catalog Processing



SHOWCAT Macro

LINK IGG0CLA0

1. Obtain virtual storage for Show Catalog work area.
2. Build VSAM catalog LOCATE interface to retrieve required data.
3. Issue SVC 26.

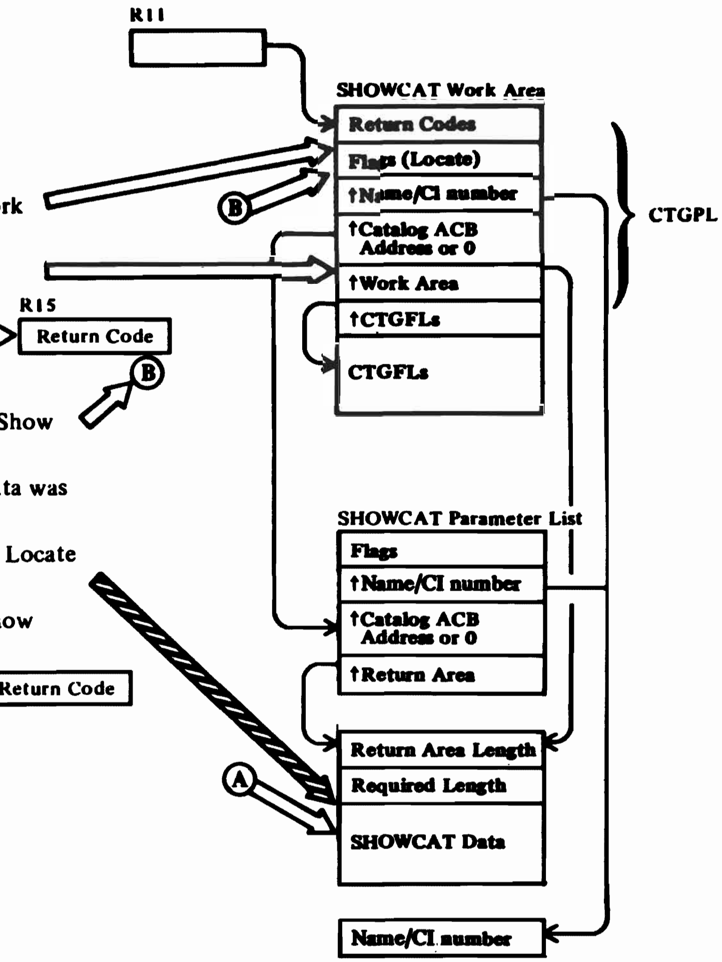
R15

Return Code

4. Translate VSAM catalog return codes into Show Catalog return codes.
5. Validity-check to ensure that the proper data was returned.
6. Transform data returned by VSAM catalog Locate into Show Catalog format.
7. Free the virtual storage obtained for the Show Catalog work area.
8. Branch register 14.

R115

SHOWCAT Return Code



Notes for Diagram EI2

The Show Catalog processor, IGG0CLA0, enables the user to obtain selected information from the VSAM catalog. This specialized user interface is mapped by macro IGGSHWPL and is invoked by the SHOWCAT macro. See *OS/VS Virtual Storage Access Method (VSAM) Options for Advanced Applications* for a complete description of the SHOWCAT macro.

The SHOWCAT macro generates the Show Catalog parameter list and issues a LINK to module IGG0CLA0. Note that this module is not a part of the VSAM supervisory load module for SVC26, IGG0CLC9.

1 The Show Catalog processor builds its conditional GETMAIN parameter list in the user-provided return area.

2 The VSAM Catalog Locate interface is built in the Show Catalog work area acquired in step 1. The CTGPL work area address points to the user return area.

3 The VSAM Catalog Locate function sets the required return area length field in the user return area and places the requested data into this return area.

4 The Show Catalog processor has equivalent error codes for VSAM catalog return codes.

5 The validity check ensures that data was actually returned and that the proper entry type is being requested.

6 The transformation causes upgrade associations and nonupgrade associations to be returned in a consistent format.

7 The Show Catalog processor always obtains a fixed amount of virtual storage for its work area. The user is responsible for providing an area of sufficient size to contain the returned data. If his area is not of sufficient size, he can use the required return area length field to obtain enough virtual storage to reissue his request.

For additional information about topics related to Show Catalog processing, see:

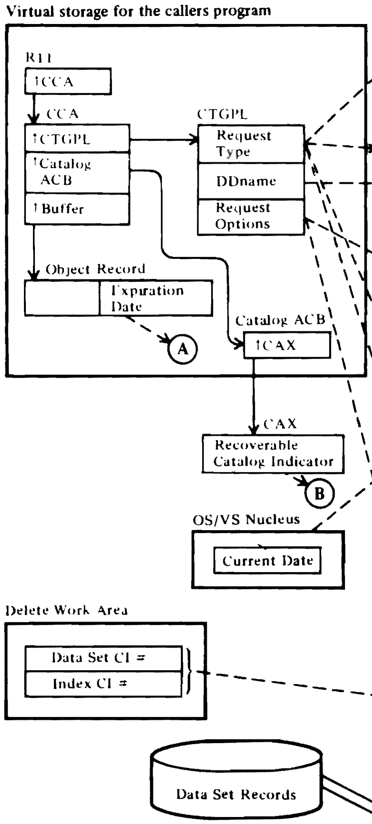
"Data Areas:"

Catalog parameter list (CTGPL) format and description
Catalog

"Diagnostic Aids:"

Catalog management return codes

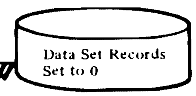
Diagram EJ1. DELETE: Remove a VSAM or NonVSAM Data Set



EB3 18

1. Did the caller specify CLUSTER, PATH, AIX, PAGESPACE, or NONVSAM?
   Yes No
2. Return to caller → CCAPROB Error Code
3. Is this a recoverable catalog?
   No Yes
4. Did the caller specify FILE?
   Yes No
5. Return to caller → CCAPROB Error Code
6. Did the caller specify PURGE?
   Yes No
7. Is the object record's expiration date value less than the current date value?
   Yes No
8. Return to caller → CCAPROB Error Code
9. Did the caller specify "TYPE (NONVSAM)"?
   No Yes → EJ2 17
10. Did the caller specify "TYPE (PATH)"?
    No Yes → EJ3 26
11. Is the data set (index) open?
    No. Yes
12. Return to caller → CCAPROB Error Code
13. Did the caller specify ERASE?
    No Yes
14. Overwrite each data control area in the user's data set with 0s.
15. Did the caller specify "TYPE (ALTERNATE INDEX)"?
    No Yes → EJ3 35
16. Did the caller specify "TYPE(CLUSTER)"?
    No Yes → EJ4 39

EJ2 17



Notes for Diagram EJ1

The DELETE command enables the user to remove from the catalog all information about a specified VSAM object or nonVSAM data set.

This figure describes the processing performed by the catalog management routines when the user issues the Access Method Services DELETE command in the form:

DELETE
(*entryname/password*)
(CATALOG (*catname/password*))
(CLUSTER | AIX | PATH | PAGESPACE |
NONVSAM)

where:

- *entryname* is the data set name of the VSAM object or nonVSAM data set to be deleted.
- *password* is the master password of the VSAM object to be deleted.
- *catname* identifies the catalog that contains the record to be deleted and specifies the catalog's password.
- CLUSTER | AIX | PATH | PAGESPACE | NONVSAM specifies the type of object to be deleted. Deletion of these types is described in this diagram. Deletion of other types is described in the following diagrams:

Diagram	Type
EK1	SPACE
EL1	USERCATALOG
EM1	ALIAS
EN1	GENERATIONDATAGROUP

The DELETE command's parameters are described in *OS/VS2 Access Method Services*.

1 IGG0CLBG: IGGPDEL

If the CATALOG parameter is not specified, the catalog record identified by the ENTRY parameter's entryname is found by a search of each catalog named by the user's JCL JOBCAT and STEPCAT DD statements, followed by a search of the VSAM master catalog. The catalog record identifier is examined to determine the record type and verify that the TYPE parameter, if specified, is correct.

Note: When deleting a catalog that has been copied onto another direct-access device (the 'source' catalog) and its 'target' catalog hasn't been deleted, IGGPDEL regards the 'source' catalog as a VSAM data set. The catalog can contain catalog records and still be deleted.

3 IGG0CLBG: IGGPDEL

If the catalog is a recoverable catalog, a DD statement must be specified for the catalog recovery area (CRA) volume.

6 IGG0CLBG: IGGPDEL

If the user specified PURGE, the data set's expiration date is ignored. See *OS/VS2 Access Method Services* for details about the RETAIN and PURGE parameters.

7 IGG0CLBG: IGGPDEL

If the user who created the data set specified the expiration date, the data set cannot be deleted until after that date (unless the PURGE parameter is specified; see step 6).

9 IGG0CLBG: IGGPDEL

If the request record type is nonVSAM, go to step 17 (EJ2). Open determination and Erase processing do not apply to nonVSAM data sets.

10 IGG0CLBG: IGGPDEL (calls **IGGPDEPT (IGG0CLB5)**)

If the request record type is PATH, go to step 26 (EJ3). Open determination, Erase, volume record, and release space processing do not apply to path catalog records.

11 IGG0CLBG: IGGPDEL (calls **IGGPDOPN (IGG0CLBG)**)

If either the data set or the index of the alternate index or cluster is already opened, the deletion of the VSAM data set will not be allowed.

13 IGG0CLBG: IGGPDEL (calls **IGGPERAS (IGG0CLBG)**)

Each of the data set's control areas is overwritten with zeros.

15 IGG0CLBG: IGGPDEL (calls **IGGPDEAX (IGG0CLB5)**)

Go to step 35 (EJ3) to explicitly delete an alternate index from the catalog. An alternate index is very similar to a key-sequenced cluster structurally, except a base cluster is always associated with an alternate index.

16 IGG0CLBG: IGGPDEL

If the request type is not a cluster, it must be a pagespace, since all other types are processed first. Pagespaces are always entry-sequenced clusters without associated paths or alternate indexes. It isn't necessary, therefore, to perform implicit delete processing for pagespaces.

Diagram EJ2. DELETE: Remove a VSAM or NonVSAM Data Set

EJ3 38 | EJ1 16 | EJ3 32

Catalog Management Buffer

Volume Catalog Record

Volume Serial Number

Data Set Directory Set of Fields (1 per Data Set (Index) in Volume)

Control Interval No. of the Object's Catalog Record

Data Space Group Set of Fields (1 per Data Space)

Sequence Number

Starting CCHH

No. Tracks Allocated to VSAM Objects

Space Map

Volume Table of Contents

Format-1 DSCB (Identifier)

Describes the VSAM Data Space on an OS/VS Data Set

Time Stamp

Catalog Management Buffer

Data Set, Index, or NonVSAM Catalog Record

CI Number of Records

Volume Information Set of Fields – Extent Descriptors (1 per Extent in the Volume)

Sequence Number of the Data Space Group

Starting CCHH Address

Ending CCHH Address

Data Set Attributes

"Unique Flag"

Update the Volume Catalog Records

Do steps 17 through 25 to process each volume information set of fields in the data set (index) catalog record:

Note: To delete space allocated to a nonVSAM data set, do steps 17 through 23 to process each volume information set of fields in the nonVSAM catalog record. Bypass steps 18, 22, 24, and 25.

17. Retrieve the volume catalog record identified by the volume serial number.

18. Delete the Data Set Directory set of fields identified by the data set (index) catalog records control interval number. (See Diagram DM1, MODIFY a Catalog Record Field's Value.")

Release the Data Set's (Index) Space

Do steps 19 through 25 to process each data space that contains part of the data set (index) records on the volume identified in step 17.

19. Does the data set (index) share its data space with other VSAM data sets or indexes?

Yes / No

20. Delete the Data Space Group set of fields identified by the volume information set of fields.

21. Is this a recoverable catalog?

Yes / No

22. Rewrite format-4 DSCB with a new current timestamp.

23. Delete the VSAM data space (See *OS/VS2 DADSM Logic*).

24. Build a list describing each of the data set's extents in the data space.

25. Modify the Space Map set of fields based on each entry in the data set's extent list to update the space map to show the tracks allocated to the data set are now unallocated.

EJ3 26

CCA

↑Buffer

Catalog Management Buffer

Volume Catalog Record

Pointer to the Data Set Directory Set of Fields "Deleted" Flag On

Catalog Management Buffer

Updated Volume Catalog Record

Volume Containing the Objects Records

Released VSAM Data Space

Available Space in a Nonunique Data Space

Notes for Diagram EJ2

17 IGG0CLA7: IGGPVMSC

IGG0CLA7: IGGPDEVG

Each volume information set of fields is retrieved from the data set (index) catalog record. If the data set (index) is a unique data set (index), the volume is mounted.

IGG0CLA7: IGGPVMSC

The volume catalog record is retrieved by forming a 44-byte true name from the volume serial number field in the volume information set of fields. The 44-byte true name for the volume catalog record is the 6-byte volume serial number followed by 38 zeros.

18 IGG0CLA7: IGGPDEDD

The volume catalog record also contains a Data Set Directory set of fields to describe each VSAM data set that is contained, partially or completely, on the volume. If the volume is a candidate volume for a data set or index, the data set or index is not described by a Data Set Directory set of fields. The Data Set Directory set of fields for a copied "source" catalog describes the copied "source" catalog cluster instead of the data set or index.

19 IGG0CLA7: IGGPVMSC

20 IGG0CLA7: IGGPDEDD

If the data set's (index's) space is not shared ("unique flag" in data set (index) attributes field on), the data space group set of fields described by the volume information set of fields (sequence number of data space group field) is deleted.

21 IGG0CLA7: IGGPVMSC

If the catalog is a recoverable catalog, the timestamp in the volume catalog record and format-4 DSCB in the volumes VTOC are not altered.

22 IGG0CLA7: IGGPDF4T

A new timestamp is obtained from the SYSTEM and the old timestamp in the volume catalog record and format-4 DSCB in the volume VTOC are rewritten with the new current timestamp.

23 IGG0CLA7: IGGPDUSC

The OS/VS DADSM Scratch routines are called by issuing SVC 29. The extents are the data space's identifier (format-1 DSCB) and extension (format-3 DSCB) are added to a format-5 DSCB. A free VTOC record (format-0 DSCB) is written over each of the data space's format-1 and format-3 DSCBs.

24 IGG0CLBF: IGGPSSCR

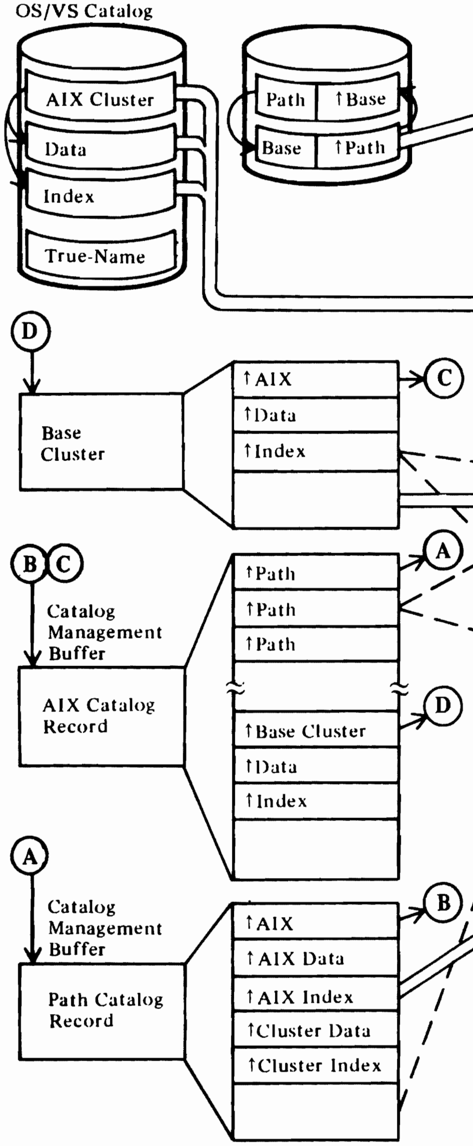
Each entry in the list identifies one of the data set's (index's) extents in one of the data spaces on the volume.

25 IGG0CLBF: IGGPSSCR

Each of the data space's extents is described in the Data Space Group set of fields.

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram EJ3. DELETE: Remove a VSAM or NonVSAM Data Set



Deletion of Path

EJ1 10

26. Retrieve the Paths base catalog record and delete the association set of fields identified by the Path's Control Interval number.

Delete CATALOG Entry Records

EJ2 25

27. Does the OS/VS system include the System Management Facility (SMF)?

No Yes

28. Write SMF record number 67 for the catalog record being deleted.

29. Rewrite the catalog record into the catalog as a free record.

30. Erase the true-name catalog record.

31. Is the object an AIX or key-sequenced cluster?

No Yes

32. Return to step 17 to process the index catalog record.

EJ2 17

33. Do step 27 through 30 for the AIX or cluster catalog record.

34. Return to the caller. Return Code

EJ1 15

Delete an AIX and Implicitly Delete Paths

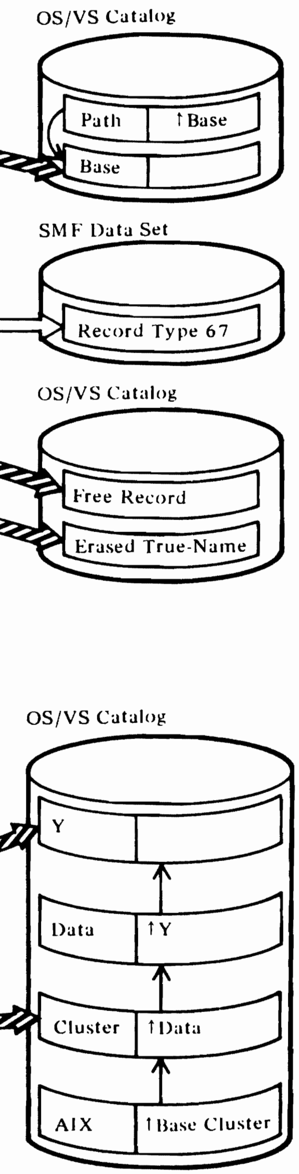
35. Retrieve each Path associated with this AIX catalog record and do steps 26 through 30 for each path.

36. Retrieve the base cluster and delete the upgrade association set of fields from the "Y" catalog record.

37. Delete the AIX association set of fields from the base cluster indicated by the AIX control interval number.

38. Go to step 17 to process the AIX catalog records.

EJ2 17



Notes for Diagram EJ3

26 IGG0CLB5: IGGPDEPT

Retrieve the path catalog record's "base" catalog record (the base catalog record can be either a cluster or an alternate index catalog record) and delete the association set of fields in the base catalog record that describes the path's control interval number. This action unchains the path catalog record from the catalog structure.

28 IGG0CLB5: IGGPDCLS

See *OS/VS2 System Programming Library: System Management Facilities* (SMF) for the format of SMF record type 67. Record type 67 is written when a VSAM cluster, path, alternate index, or nonVSAM data set defined in a VSAM catalog is deleted.

30 IGG0CLB5: IGGPDCLS

The DELETE routines erase the data set's true-name record and delete all references to the data set's DSNAME in the catalog's index.

Note: True-name records do not exist for the data set or index record of a copied "SOURCE" catalog.

31 IGG0CLB5: IGGPDCLS

If the object catalog record type is an alternate index or a key-sequenced cluster, steps 17 through 30 are performed to delete the index catalog record.

33 IGG0CLB5: IGGPDCLS

Steps 27 through 30 are performed for the cluster or alternate index catalog records. All RACF profiles associated with the cluster's components are deleted by IGGPROBD (IGG0CLDD).

35 IGG0CLB5: IGGPDEAX (calls **IGGPDIPT (IGG0CLB5)**)

When deleting an alternate index, all paths associated with the alternate index are implicitly deleted first. This process is performed by retrieving each path record and completing steps 26 through 30.

If a RACF-indicated path is deleted, IGGPRPTH (IGG0CLDD) deletes the path's profile.

36 IGG0CLB5: IGGPDUPG (calls **IGGPUDEL (IGG0CLB1)**)

The alternate index base cluster is retrieved and the upgrade association set of fields described by the alternate index's data set (index) control interval numbers is deleted from the "Y" catalog record associated with the cluster's data set.

37 IGG0CLB5: IGGPDEAX

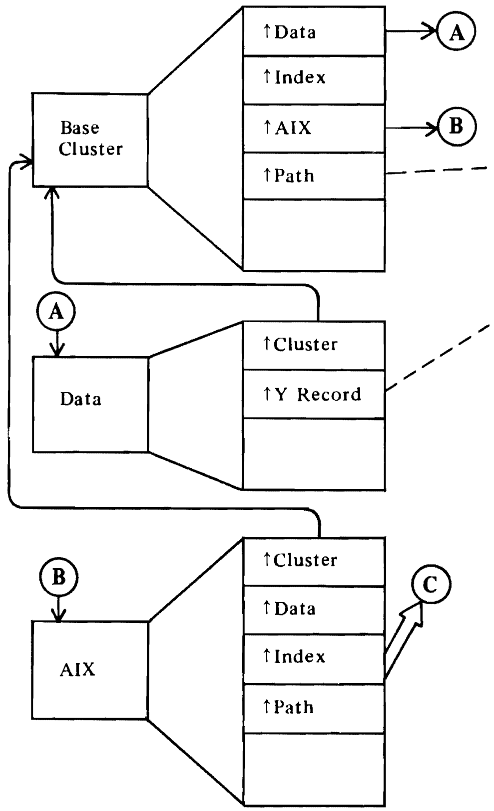
Delete the association set of fields in the base catalog record that describes the alternate index control interval number. This action unchains the alternate index catalog record from the catalog.

38 IGG0CLB5: IGGPDEAX

To process the alternate index, data set, and index catalog records, steps 17 through 34 are performed.

Page of SY26-3826-3
Revised July 28, 1978
By TNL SN26-0912

Diagram EJ4. DELETE: Remove a VSAM or NonVSAM Data Set



EJ1 16

Delete a Cluster and Implicity Delete AIXs and Paths

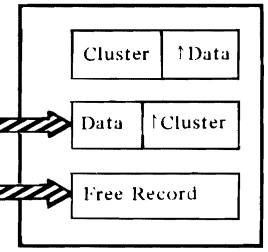
39. Retrieve each AIX record associated with the base cluster and do step 11 and steps 35 through 37. Then do steps 17 through 33 (skipping step 26) for each AIX.

40. Retrieve each path record associated with the base cluster and do steps 26 through 30 for each path

41. Delete the cluster's associated "Y" catalog record.

42. Go to step 17 to process the cluster catalog record.

EJ2 17



Notes for Diagram EJ4

39 IGG0CLB5: IGGPDIAX (calls **IGGPDIPT (IGG0CLB5)**)

When deleting a cluster, all associated alternate index paths are implicitly deleted first, followed by all the associated alternate indexes and the associated cluster paths.

Each alternate index record is retrieved and step 11 is performed to assure that the data sets (index) are not opened. Steps 35 through 37, 17 through 25, and 27 through 33 are performed to implicitly delete all associated alternate indexes and alternate index paths.

40 IGG0CLBG: IGGPDEL (calls **IGGPDIPT (IGG0CLB5)**)

Retrieve each cluster's associated path record and perform steps 26 through 30 for each path.

41 IGG0CLBG: IGGPDEL (calls **IGGPDUPG (IGG0CLB5)**)

The "Y" catalog record associated with the cluster data set record and the related association set of fields in the data set record are deleted.

42 IGG0CLBG: IGGPDEL (calls **IGGPDCLS (IGG0CLB5)**)

To process the cluster, data set, and index catalog records, steps 17 through 34 (skip step 26) are performed.

For additional information about topies related to DELETE processing, see:

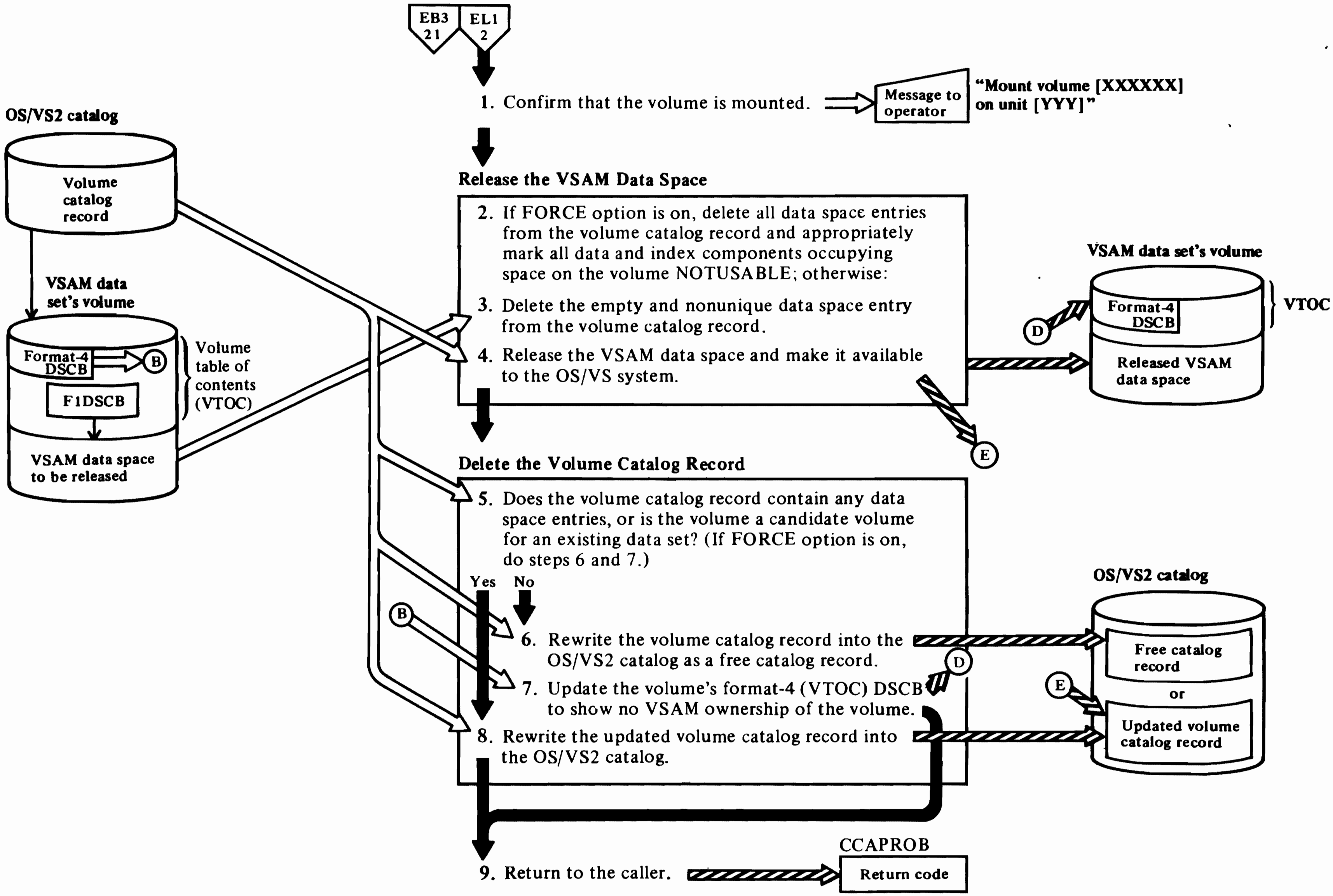
"Data Areas:"

Volume catalog record description
Data set directory set of fields description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram EK1. DELETE SPACE: Release All of the Empty VSAM Data Spaces on a Volume



Notes for Diagram EK1

The DELETE SPACE command enables the user to release all VSAM data spaces on a specified volume. This figure describes the processes performed by the catalog management services DELETE SPACE routine when the user issues the Access Method Services DELETE SPACE command in the form:

DELETE
(entryname/password) SPACE
[CATALOG (catname/password)]
[FILE (dname)] [FORCE]

where:

- *entryname* is the volume serial number of a direct access volume containing VSAM data spaces to be deleted.
- *catname* is the name of the catalog that contains the volume's catalog record.
- *password* is the catalog's master, control interval, or update password.
- *FILE* identifies the JCL statement that causes the volume to be mounted.

The DELETE command parameters are described in *OS/VS2 Access Method Services.*

1 IGG0CLBL: IGGPDELS and **IGGPDLVM**

If the volume isn't already mounted and available for use, the DELETE SPACE routine issues the appropriate mount message to the operator.

2 IGG0CLAI: IGGPFDSP, IGGPDFMI

FORCE DELETE uses DADSM SCRATCH to release all VSAM data space and make it available to other OS/VS system users.

3 IGG0CLBL: IGGPDLSH, IGGPDLSD, and **IGGPDLCB**

The volume catalog record contains a Data Space Group set of fields to describe each VSAM data space on the volume.

See "Data Areas" for details about the volume catalog record and its Data Space Group sets of fields.

4 IGG0CLBL: IGGPDLSC

The OS/VS DADSM Delete routine releases the empty nonunique VSAM data space and makes its space available to other OS/VS system users.

See *OS/VS2 DADSM Logic* for details about deleting an OS/VS data set (to DADSM, the same as a VSAM data space).

5 IGG0CLBL: IGGPDELS

When the volume is totally empty, the volume catalog record can be deleted from the catalog. This occurs when there are no Data Space Group sets of fields and no Data Set Directory Entry sets of fields in the volume catalog record.

6 IGG0CLBL: IGGPDLET

IGGPDLET rewrites the record as a free catalog record so that it is available for future assignment.

7 IGG0CLBL: IGGPDLET

The format-4 (VTOC) DSCB is the first entry in a direct-access volume's VTOC. It contains the volume's owner's identification and information on how the volume is used.

See *OS/VS2 Data Areas* for DSCB details.

8 IGG0CLBL: IGGPDLET

For additional information about topics related to DELETE SPACE processing, see:

"Data Areas:"

Volume catalog record description
Data set group set of fields description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram EL1. DELETE CATALOG: Release an OS/VS2 Catalog

EB3
21

1. Is this a FORCE DELETE request?

No Yes

2. Release all data spaces. (See Diagram EK1, "DELETE SPACE: Release all of the Empty Data Spaces on a Volume.")

3. Is this a DELETE MASTERCATALOG request?

No Yes

4. If PURGE is not on and current date is less than expiration date, return to the caller.

CCAPROB
Error code

5. Is FORCE option on?

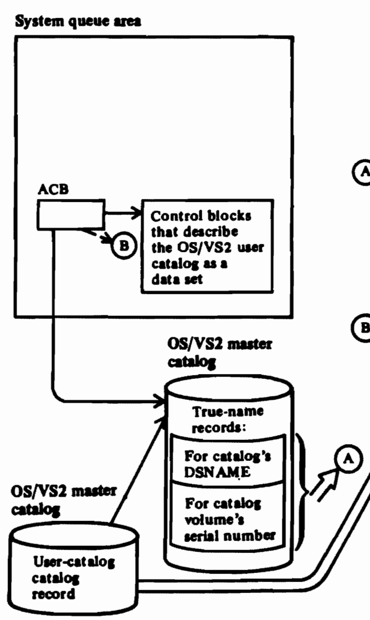
Yes No

6. Is the catalog empty?

Yes No

7. Return to caller

CCAPROB
Error code



Release a Catalog's Data Spaces

8. Release all empty data spaces. (See Diagram EK1, "DELETE SPACE: Release All of the Empty Data Spaces on a Volume.")

Volume that contained the OS/VS2 catalog
Freespace

Delete a VSAM User Catalog

9. Close the OS/VS2 user catalog's ACB.

10. Delete the user-catalog catalog record from the OS/VS2 master catalog.

OS/VS2 Master Catalog
Free catalog record

11. Return to the caller.

CCAPROB
Return code

Notes for Diagram EL1

The DELETE USERCATALOG command enables the user to release a catalog's space and make the space available to other OS/VS system users. The catalog must be empty (see step 6 notes) or the request is rejected. This figure describes the processes performed by the catalog management services DELETE CATALOG routines when the user issues the DELETE command in the following form:

DELETE
(entryname/password)
USERCATALOG [FORCE]

where:

- *entryname* is the DSNAME of the catalog to be deleted.
- *password* is the user catalog's master password.

The DELETE command parameters are described in *OS/VS2 Access Method Services*.

6 IGG0CLAF: IGGPDELC

If the catalog contains more than two true-name catalog records, it is not empty and cannot be deleted, unless the FORCE option is on.

Note: When a catalog that was copied onto another direct-access device (the 'TARGET' catalog) is being deleted, and its 'SOURCE' catalog hasn't been deleted, IGGPDELC allows the TARGET catalog to be deleted even though it isn't empty. See the discussion on copying a catalog in the "Introduction".

8 IGG0CLAF: IGGPSDSP

The volume catalog record contains an entry for each VSAM data space allocated on the volume. Each entry contains the data necessary to free the data space.

Diagram EK1, DELETE SPACE, shows how each VSAM data space is released and it's space made available to other OS/VS system users.

9 IGG0CLAF: IGGPDELC

The user catalog is described, as a data set, by its ACB. See VSAM Close Processing, in *OS/VS2 VSAM Logic*, for a description of closing a VSAM data set.

For additional information about topics related to DELETE CATALOG processing, see:

"Data Areas:"

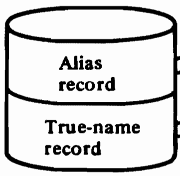
Volume catalog record
Access method control block (ACB) description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram EM1. DELETE ALIAS: Remove a Non VSAM Data Set's or User Catalog's Alternate DSNAME

OS/VS2 catalog



EB3
21

1. Retrieve the alias catalog record.
2. Adjust the alias chain pointers in the previous and next alias catalog records.
3. Delete the true-name catalog record that points to the alias catalog record.
4. Delete the alias catalog record.
5. Return to the caller.

CCAPROB

Return code

OS/VS2 catalog

Notes for Diagram EM1

The DELETE ALIAS command is used to remove a nonVSAM data set's or user catalog's alternate DSNAME. In addition, DELETE ALIAS processing adjusts the alias catalog record chain (previous and next alias record pointers in association sets of fields) to account for the deleted record. This diagram describes the processes performed by the catalog management services DELETE ALIAS routines when the user issues the DELETE command in the following form:

DELETE
ALIAS
entryname
[CATALOG (catname/password)
  CLUSTER | SPACE | USERCATALOG |
  NONVSAM | ALIAS | GDGBASE]

where:

- *entryname* is the name of the alias to be deleted.
- *password* is the update (or higher) level password of the catalog containing the alias to be deleted.
- *CATALOG* identifies the catalog that contains the record to be deleted, and specifies the catalog's password.

1 IGG0CLA5: IGGPDELX

IGGPDELX issues the GETREC macro instruction to retrieve the alias catalog record.

3 IGG0CLA5: IGGPDELX

IGGPDELX calls IGGPCCFM (IGG0CLA5) to build a CTGPL and CTGFLs to update the previous and next record's pointers (part of the Alias record chain) to the record being deleted.

For additional information about topics related to DELETE ALIAS processing, see:

"Data Areas:"

Alias catalog record
Catalog parameter list (CTGPL) description and format
Field parameter list (CTGFL) description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram EN1. DELETE GENERATION DATA GROUP: Remove a Generation Data Group

EB3
21

Check the Generation Data Group's Expiration Date

1. Is the PURGE option specified by the user?
   Yes No
2. Is the generation data group's expiration date value less than the current timestamp value?
   Yes No
3. Return to the caller. → CCAPROB: Error code

OS/VS2 catalog: GDG base record; True-name record

4. Retrieve the generation data group (GDG) base catalog record.
5. Have all generation data set's in the group been deleted?
   Yes No
6. Return to the caller. → CCAPROB: Error code
7. Delete the true-name catalog record that points to the GDG base catalog record.
8. Delete the GDG base catalog record.
9. Return to the caller. → CCAPROB: Return code

OS/VS2 catalog: Free catalog records

Notes for Diagram EN1

The DELETE GENERATIONDATAGROUP command is used to remove a generation data group's base record after all generaton data sets in the group have been deleted. This diagram describes the processes performed by the catalog management services DELETE GENERATIONDATAGROUP routines when the user issues the DELETE command in the following form:

DELETE
GENERATIONDATAGROUP (*entryname*)
CATALOG (*catname/password*)
(PURGE) (FORCE)

where:

- *entryname* is the name of the generation data group to be deleted.
- *password* is the update (or higher) level password of the catalog containing the generation data group to be deleted.
- *catname* identifies the catalog that contains the record to be deleted, and specifies the catalog's password.

1 IGG0CLBG: IGGPDEL

If the user specified PURGE or FORCE, the generation data group's expiration date is ignored. See *OS/VS2 Access Method Services* for details about the PURGE parameter.

2 IGG0CLBG: IGGPDEL

If the definer of the generation data group specified an expiration date, the generation data group cannot be deleted until after that date (unless PURGE or FORCE is specified—see step 1).

4 IGG0CLA5: IGGPDELB

6 IGG0CLA5: IGGPDELB

If all of the generation data sets in the groups haven't been deleted and the FORCE option is not on, IGGPDELB sets error return code RCDNECAT and returns to the caller.

9 IGG0CLA5: IGGPDELB

IGGPDELB issues the DELREC macro instruction to delete the generation data group base catalog record and its true-name catalog record.

For additional information about topies related to GENERATIONDATAGROUP processing, see:

"Data Areas:"

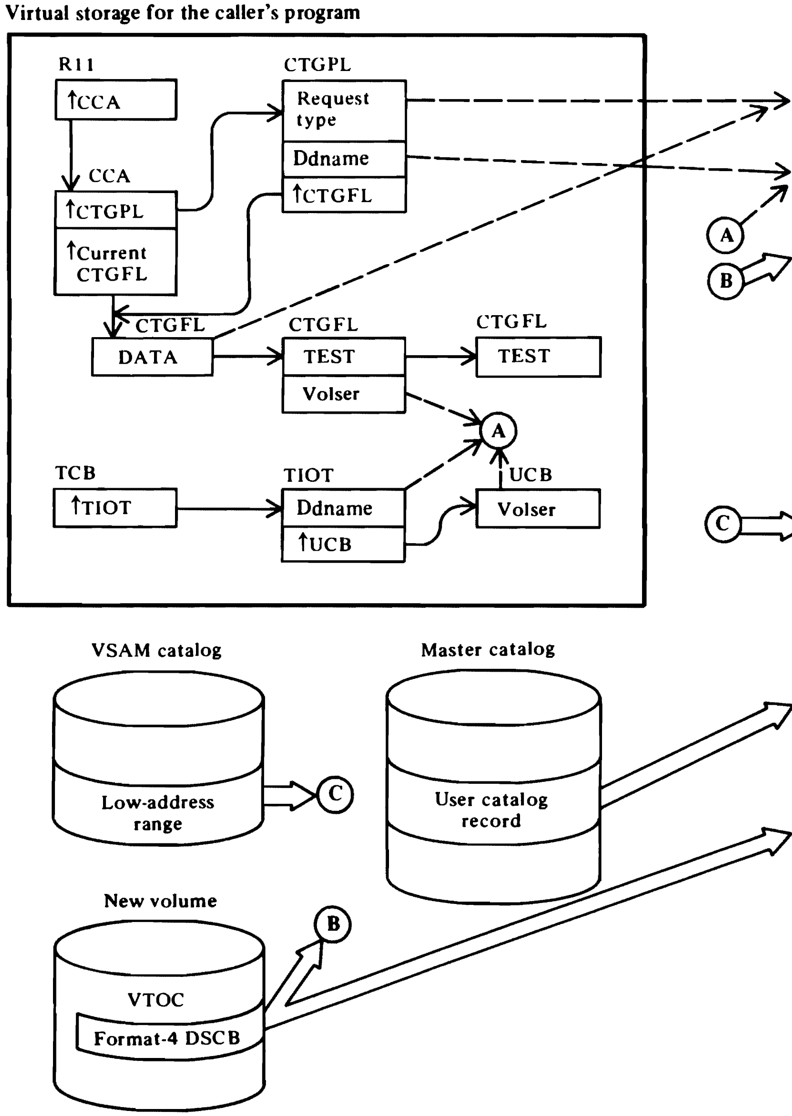
GDG Base catalog record
True Name catalog record
Catalog parameter list (CTGPL) description and format

"Diagnostic Aids:"

Catalog management return codes

Diagram E01. CONVERTV: Convert a Volume to or from Mass Storage

EB3
21



1. Verify the caller's authority to convert the volume.
2. Check the validity of the catalog parameter list and the field parameter lists.
3. Verify that the ddname is correct and the volume is mounted.
4. Indicate the volume is unusable for the duration of conversion.

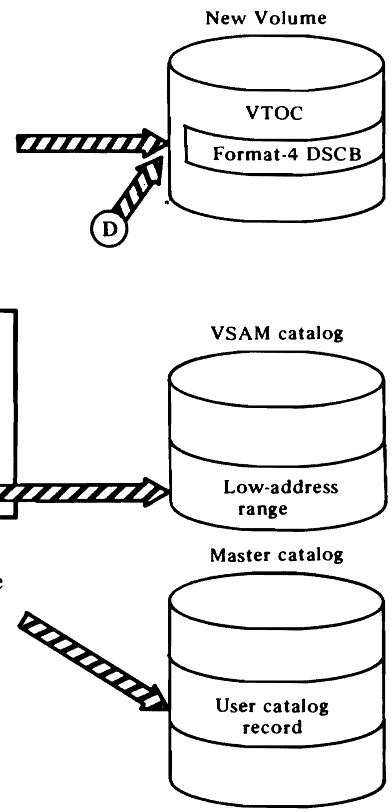
Convert volume infromation in the catalog that owns the volume

C

Repeat steps 5 and 6 until all catalog entries that indicate the converted volume have been processed:

5. Retrieve a catalog record.
6. If the catalog record points to the volume, change the device-type indication.

7. If a user catalog is on the converted volume, change the pointer to the user catalog to indicate the new device type.
8. Indicate the volume is usable. D



Notes for Diagram EO1

The CONVERTV command enables the user to convert the contents of a direct-access storage volume to a mass storage volume, or vice versa, and to have catalog information that indicates the old device type changed to indicate the new device type.

This figure describes the processing performed by the catalog management services routines when the user issues the Access Method Services CONVERTV command in the form:

CONVERTV
FROMFILE (ddname)
TOFILE (ddname)
(RECATALOG (ALL VSAMCATALOG))
(CATALOG (catname (/password)))

where:

- FROMFILE and TOFILE identify the DD statements that cause allocation of the devices from and to which the volume's contents are being converted.
- RECATALOG indicates that a VSAM catalog owns the volume being converted and indicates the extent of recataloging that is to be done.
- CATALOG identifies the user catalog, if there is one, on the volume being converted.

CONVERTV is described in *OS/VS Mass Storage System (MSS) Services for Space Management.*

1 **IGG0CLBZ: IGGPCONV**

Issues the TESTAUTH macro, which checks whether the calling program is an APF (authorized program function).

2 **IGG0CLBZ: IGGPVALI**

3 **IGG0CLBZ: IGGPCONV**

4 **IGG0CLBU: IGGPF4RD**

Reads the format-4 DSCB.

IGG0CLBU: IGGPF4WR

Writes the format-4 DSCB. If CONVERTV fails and a user subsequently attempts to open a data set on the volume, he will receive an error code from Open.

Before setps 5 and 6:

IGG0CLAG: IGGPPUPC

Writes the modified volume record. The identity of this record is passed to IGG0CLBZ from Access Method Services.

IGG0CLAG: IGGPRCCR

Updates the catalog control record in the catalog that owns the volume to indicate the next free control interval.

5 **IGG0CLBZ: IGGPGALO**

6 **IGG0CLAV: IGGPMOD**

Modifies device-type fields in catalog records that point to the volume being converted.

7 **IGG0CLAH: IGGPSCAT**

Searches the master catalog for the user catalog record.

IGG0CLAV: IGGPMOD

Writes the modified master catalog record that points to the user catalog. In step 5, IGG0CLBZ discovers that a user catalog is on the volume being converted when it finds a user catalog record that indicates the volume's volume serial number.

IGG0CLA3: IGGPRPLF

Releases the master catalog from exclusive control.

8 (See step 4.)

PROGRAM ORGANIZATION

This chapter discusses the implementation of the catalog management routines, showing how the actual modules and procedures fit together.

The final authority of any program is the assembly or compilation listing. These module discussions complement the assembly listing comments, and assume that the listings are at hand. You should have them available for any in-depth analysis.

Catalog management program listings are the key to the catalog management program logic. You get into the listings from the method of operation diagrams. Once you have located the module or routine name that interests you in the diagrams, you are ready to turn to the listing to find the additional information you require.

Module Prologues

Each catalog management module listing begins with a description of the module, called the module prologue.

The information contained in catalog management module prologues is described in the topics that follow.

Module name: The external procedure name of the module (for example, IGG0CLA1).

Descriptive name: The English name of the module (for example, Catalog Management: Initial Processing).

Status: The version and release level of the module.

Function: A brief step-by-step explanation of the functions performed by this module. Function is divided into steps so that you may more easily locate the routine responsible for each step.

Notes: A generalized heading that includes (1) any dependencies, for example, CPU model or features, that will affect the operation of this module, (2) any restrictions that apply to this module, (3) symbols used to represent registers and register usage, (4) symbolic name of the maintenance area for this module and whether the maintenance area is used or reserved, and (5) any special terms and acronyms that are used within this module that are not necessarily used elsewhere in the documentation.

Module type: A description of the type of this module (for example, procedure or macro), the name of the compiler used/required to create this module, the amount of storage required by this module for executable code and associated data, and the attributes of the module (for example, reentrant or read-only).

Entry point: The name of the point at which control can enter this module, the conditions of entry, the calling sequence by which control was given, including any parameters passed and the names of modules that can enter at this entry point.

Input: A description of anything this module gets or references, for example, registers, control blocks, and data. The means by which this module gains access to the input is included.

Output: A description of registers, control blocks, and data areas at output; any messages issued as a result of this module's processing are included.

Exit-normal: A description of conditions at and reasons for normal exit from this module, including the names of modules called by this module.

Exit-error: A description of conditions at and reasons for any error exit from this module.

External references: A list of modules, data areas, etc., defined outside of or accessible outside of this module.

Tables: A list of all local tables and work areas, that is, data areas built and used only within this module.

Macros: A description of system macros used by this module.

Change activity: A list of any change activity to this module.

Module Flow Compendiums

A compendium and its notes describe the flow of control among procedures and modules to perform a function. The compendium is a supplement to the function's method of operation diagram.

The compendium's notes describe how each procedure and module contribute to the completion of the function, and under what circumstances the procedure or module is called.

Reading Module Flow Compendiums

Program organization compendiums are descriptions of catalog management functions, in terms of module (procedure) calls and usage.

The compendium shows the flow of control among catalog management modules in order to perform a function. Figure 6 shows a compendium figure. A single-headed arrow indicates that control is passed from one module to another and does not return. A double-headed arrow indicates that control is returned when the "called" module completes its processing.

Blocks that are indented (otherwise contained within another block) are called to perform a specified function and return, when finished, to the caller of the routine.

Numbers and letters in **bold-face type** refer to descriptive notes. The notes tell what the caller expects the called module (procedure) to do.

Catalog management procedures call certain procedures so frequently that, if each call were shown, the catalog management compendiums would be cluttered. For this reason, whenever a procedure calls one of these (frequently called) procedures, its module identifier (last two letters of the module name: 'xx' in IGG0CLxx) is listed instead of drawing a separate block to show the procedure call. The frequently-called modules are:

- AG: IGG0CLAG—Catalog Management Input/Output Procedures
- AV: IGG0CLAV: IGGPMOD—Modify Catalog Field(s)
- AZ: IGG0CLAZ: IGGPEXT—Extract Catalog Field(s)
- BI: IGG0CLBI—Catalog Management Input/Output Procedures
- BV: IGG0CLBV—SMF Record Processing Procedures
- B3: IGG0CLB3—SMF Record Processing Procedures

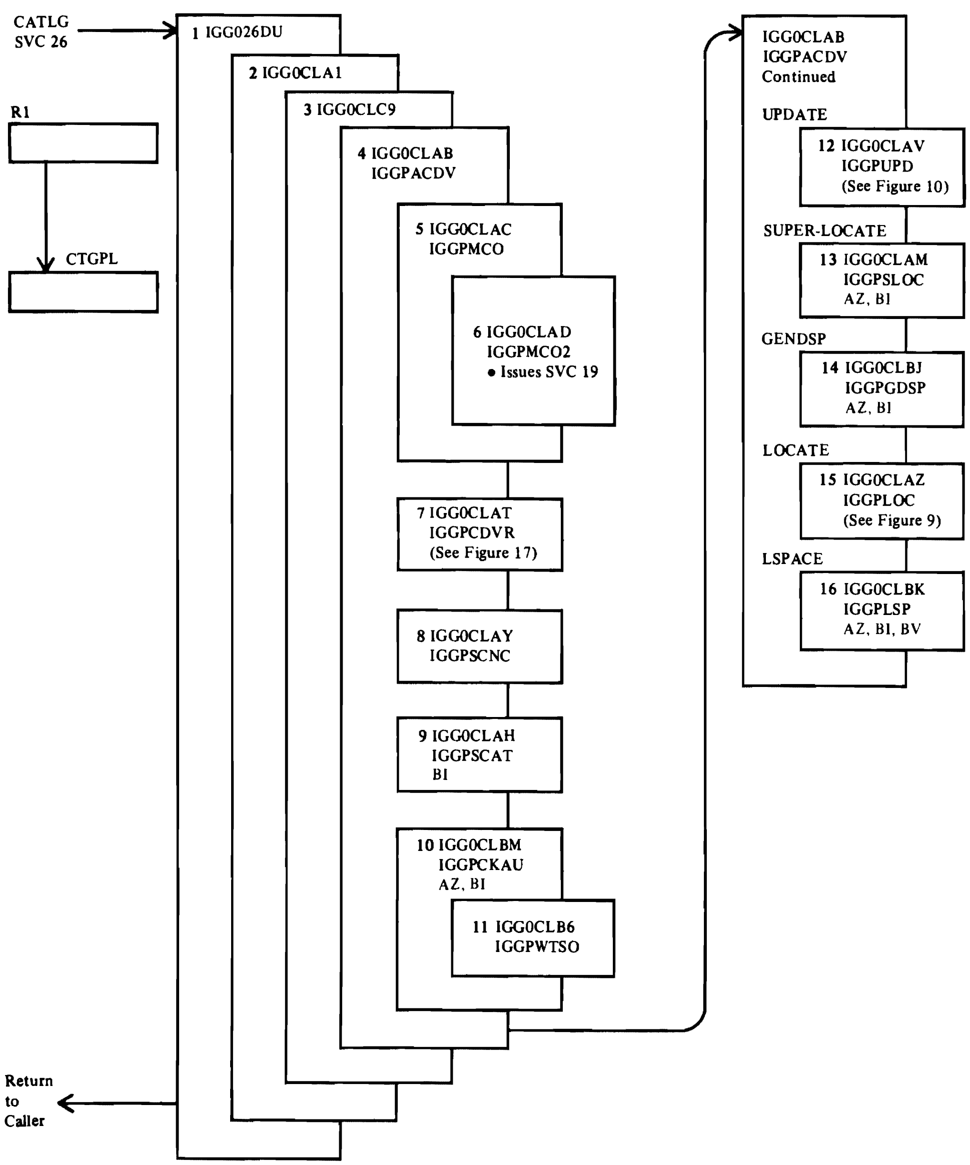


Figure 6. Program Organization Compendium Figure

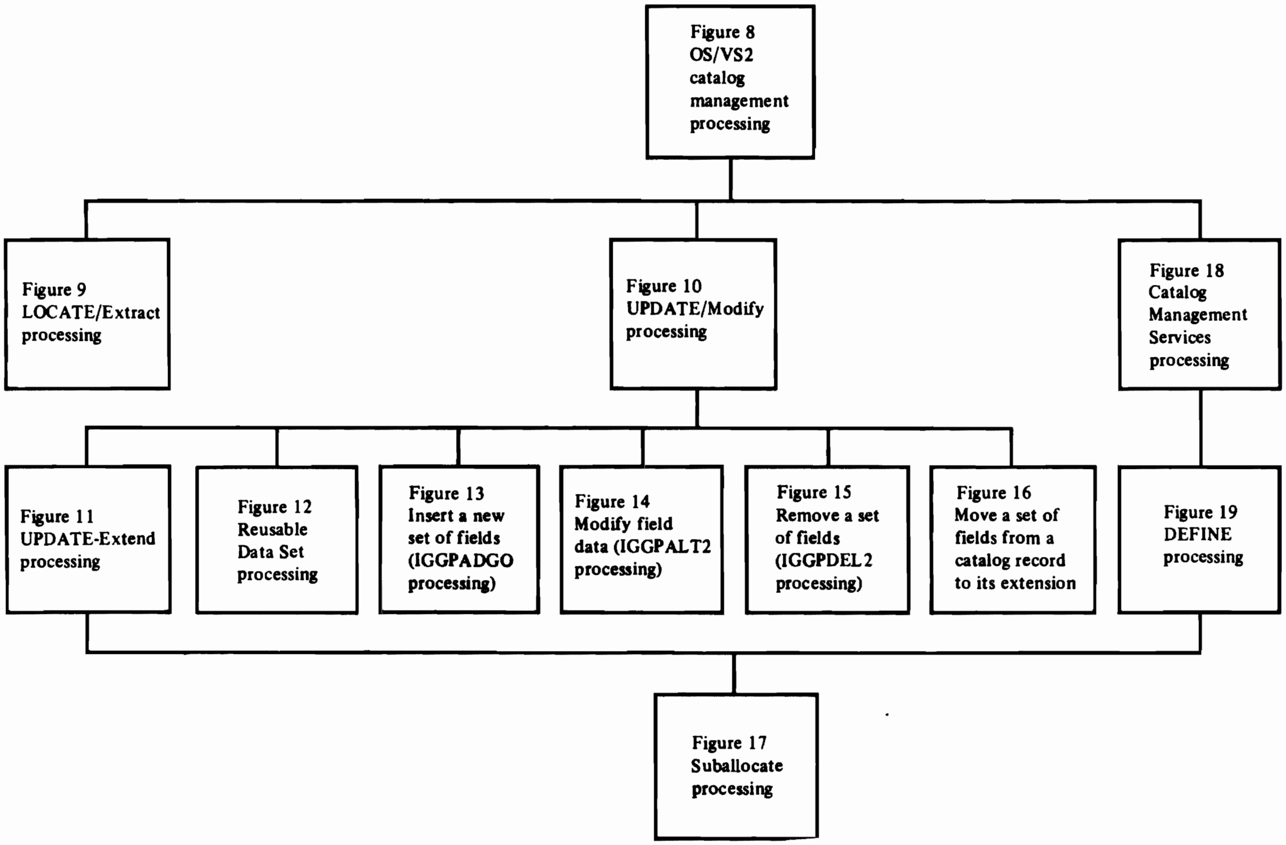


Figure 7. Catalog Management Program Organization Contents

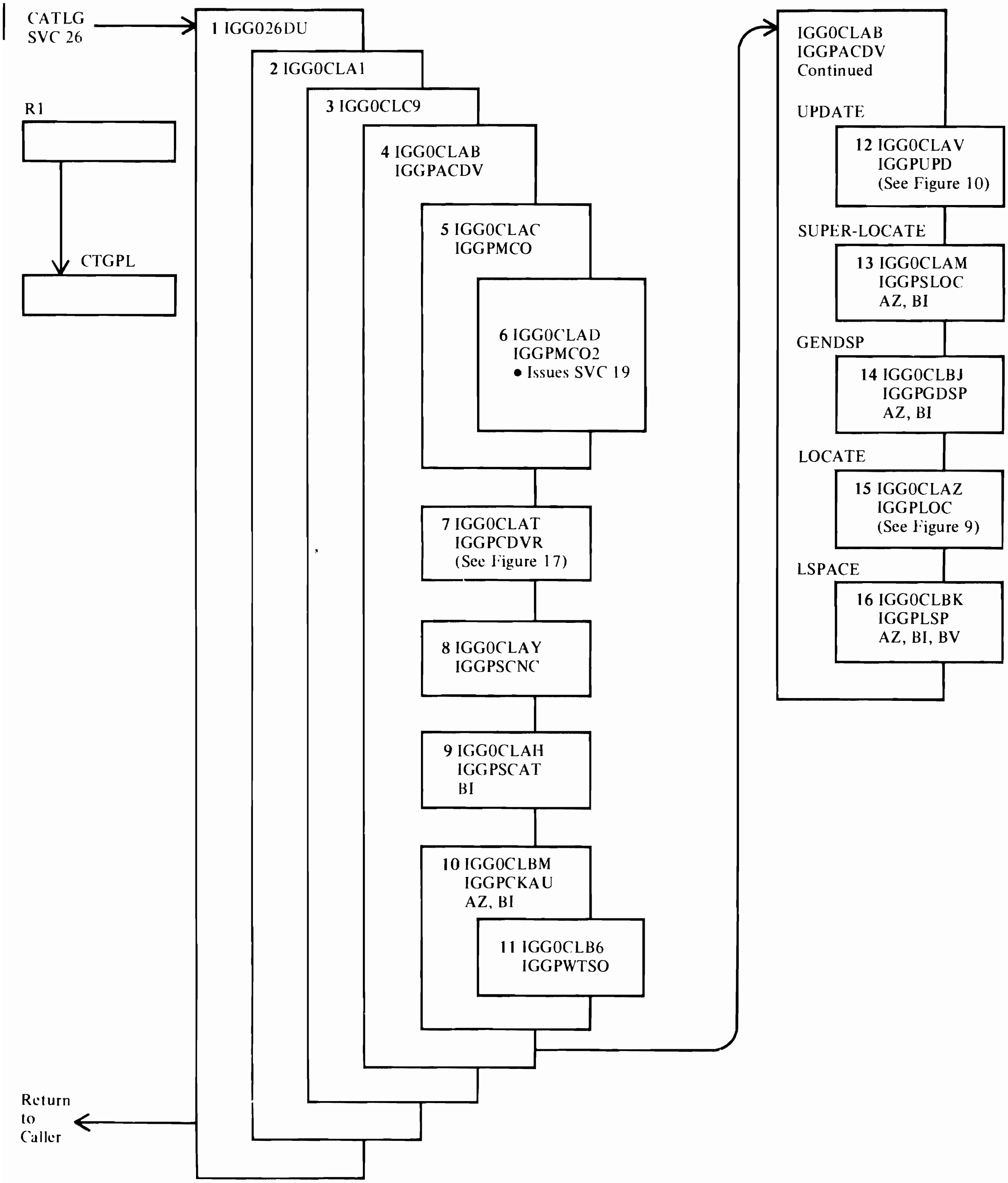


Figure 8. OS/VS2 Catalog Management Processing

Notes for Figure 8

This figure describes the overall processing of OS/VS2 Catalog Management in terms of flow-of-control between catalog management procedures. Detailed descriptions of LOCATE, UPDATE, and Catalog Management Services processing are on following compendiums.

1 IGC0260U is an OS/VS2 Catalog Management module.

2 IGG0CLA1 is the OS/VS2 Catalog Management load module.

3 IGG0CLC9 builds the CCA for the request and performs initial and final OS/VS2 Catalog Management processing.

4 IGGPACDV is the OS/VS2 Catalog Management Common Processing procedure.

5 When the master catalog is not open, IGGPMCO is called to open it.

6 IGGPMCO and IGGPMCO2 initialize an ACB to describe the master catalog, then issue SVC 19 to open it.

7 When the CTGPL indicates a Catalog Management Services request (DEFINE, ALTER, DELETE, or LISTCAT), the Catalog Management Services: Common Processing procedure (IGGPCDVR) is called.

8 IGGPSCNC checks and initializes the CTGFLs for the other types of Catalog Management requests (LOCATE and UPDATE).

9 IGGPSCAT retrieves the catalog record identified by the CTGPL (the object's base catalog record). Extensions to the base record are retrieved as they are needed.

BI: IGGPGET issues GET to retrieve catalog records.

10 IGGPCKAU verifies the caller's authorization to perform the CTGPL's request.

AZ: IGGPEXT locates the password information required by IGGPCKAU.

BI: IGGPGET issues GET to retrieve the object's catalog record that contains its password set of fields (group occurrence).

11 When the user is on a TSO terminal, IGGPWTSO issues requests to the TSO terminal for the required password.

12 When the caller's request is UPDATE, IGGPUPD receives control. The caller may request that his VSAM data set be extended (IGGPUPDE), that it be reset (IGGPRUS), or that it be updated (IGGPUPD). If the request is for an update, only fixed-length record fields should be changed.

13 When the caller's request is SUPERLOCATE, IGGPSLOC processes it. SUPERLOCATE is issued to obtain the volume serial number of each volume that contains a part of the caller-specified cluster's data set or index.

AZ: IGGPEXT locates the volume information sets of fields (group occurrences).

BI: IGGPGET issues GET to retrieve catalog records as required.

14 The caller's GENDSP request is processed by either IGGPGDSP (request for a nonunique data space) or IGGPGUDS (request for a unique data space). GENDSP obtains the control interval numbers of the catalog record(s) of each object (cluster, data set, index, and catalog) contained in a VSAM space identified by a DSNAME.

AZ: IGGPEXT locates the data space group set of fields (group occurrence) that describes the data space and the data set directory entry sets of fields that point to the catalog records of VSAM objects in the volume's data space.

BI: IGGPGET issues GET to retrieve volume catalog records as required.

15 When the caller's request is LOCATE, IGGPLOC processes it. The caller is allowed to retrieve fixed-length and (entire) variable-length catalog record fields.

16 When the caller's request is LSPACE, IGGPLSP processes it.

AZ: IGGPEXT locates the data space group sets of fields (group occurrences) that describe each shared (nonunique) data space on the volume.

BI: IGGPGET issues GET to retrieve volume catalog records as required.

BV: IGGPSMFL writes SMF record type 69—VSAM Data Space Defined or Deleted.

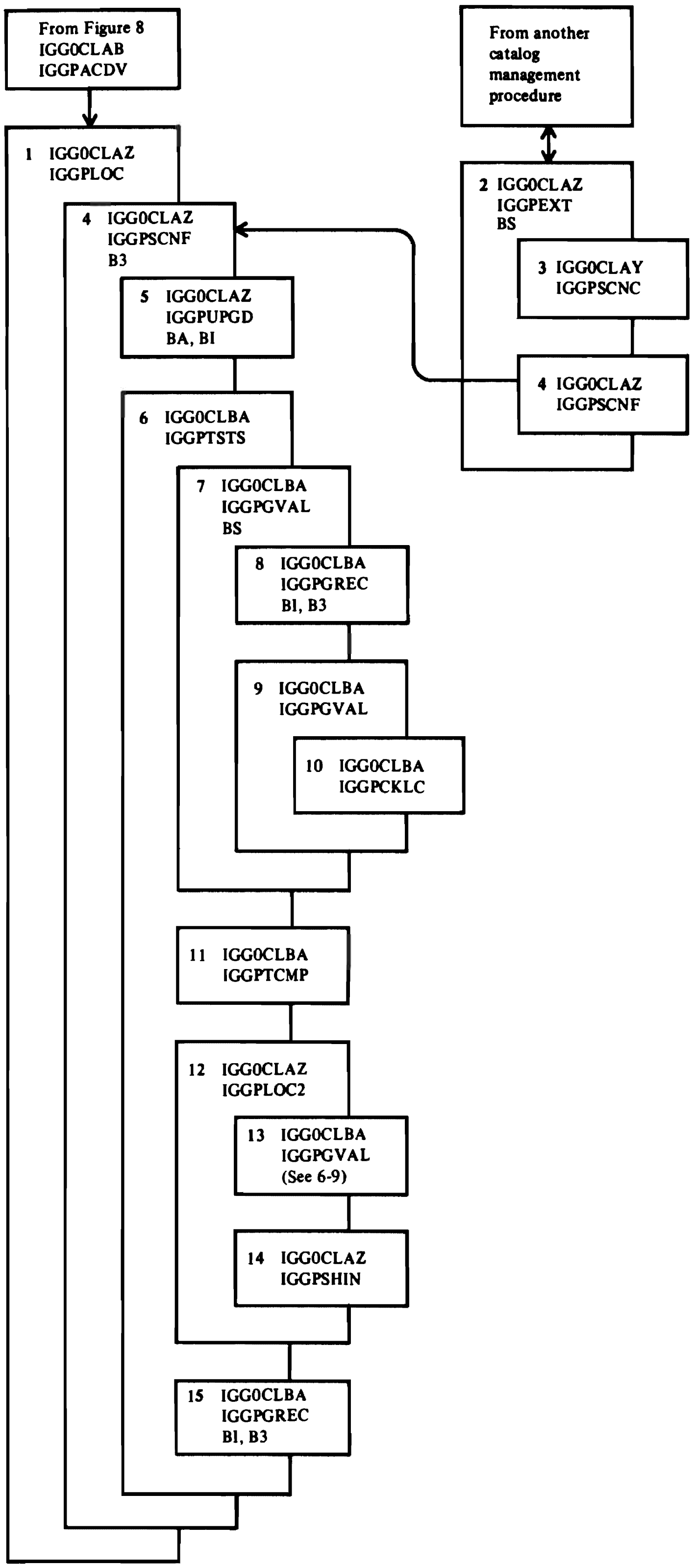


Figure 9. LOCATE/Extract Processing

Notes for Figure 9

This figure describes the processing done when a caller wants to retrieve a catalog record's field. When the caller is another part of catalog management, the caller is allowed to retrieve any fixed-length or varaible-length field, or a part of a variable-length field. When the caller is an OS/VS component that issued the LOCATE command, the caller is allowed to retrieve only complete fixed-length and variable-length fields.

1 IGGPLOC retrieves an entire (fixed-length or variable-length) catalog record field's contents for the caller (other than an internal OS/VS2 catalog management procedure).

2 IGGPEXT retrieves a fixed-length, variable-length, or part of a variable-length catalog record field's contents for the caller (an OS/VS2 catalog management routine).

If the specified field name indicates a "derived information" field, IGG0CLBS processes the field name.

3 IGGPSCNC initializes CTGFLs with dictionary information required to find the field in the catalog record, and ensures that the CTGFLs are valid.

4 IGGPSCNF (**6** through **13**) processes each CTGFL addressed by the caller's CTGPL to retrieve all catalog information that satisfies the caller's request.

B3: IGGPSMFG makes a copy of the base catalog record in case it is updated later by a modify call.

5 IGGPUPGD retrieves the associated upgrade entry record if this is an upgrade field name.

BA: IGGPGVAL retrieves the connecting association to the upgrade entry from the current base entry.

BI: IGGPGET issues GET to retrieve connecting entries to the upgrade entry, as well as the upgrade entry itself.

6 When the CTGFL (addressed by the caller's CTGPL) addresses CTGFLs-for-tests, IGGPTSTS (**6** through **10**) processes each CTGFL-for-tests to identify each set-of-fields (group occurrence) that satisfies the test conditions.

7 IGGPGVAL retrieves one catalog-record-field's value.

If the specified field name indicates a "derived information" field, IGG0CLB5 processes the field name.

8 If more set-of-fields pointers (group occurrence pointers) are in an extension of the base catalog record, or

If the specified set-of-fields pointer (group occurrence pointer) contains the control interval number of an extension record, IGGPGREC retrieves the required extension record.

BI: IGGPGET issues GET to retrieve the catalog record.

B3: IGGPSMFG makes a copy of the catalog record in case it is updated later.

9 IGGPLVAL locates the field within the catalog record.

10 IGGPCKLC verifies that the field exists (ie. the requested field is in the catalog record or one of its sets of fields (group occurrences)).

11 IGGPTCMP compares the catalog record field's value to the caller's test data and, if the compare is OK, saves the sequence number of the catalog record field's set-of-fields pointer (group occurrence pointer).

12 IGGPLOC2 retrieves catalog-record-field contents to satisfy the caller's request. If the caller's CTGFL specifies a special field (one not in the catalog record) or a combination field-name (a field-name that identifies a group of related fields), IGGPLOC2 processes the field-name and calls IGGPGVAL, as required, to retrieve the requested information.

If the caller provided CTGFLs-for-tests, each catalog record field is retrieved if it is:

- Identified by the CTGFL's (addressed by the CTGPL) field name, and
- Contained in a set of fields (group occurrence) that satisfies all tests associated with the CTGFL (addressed by the CTGPL). The set-of-fields pointer's (group occurrence pointer's) sequence number is set by step 11.

If the caller didn't provide CTGFLs-for-tests, the contents of each catalog record field identified by the CTGFL's field name is retrieved.

13 IGGPGVAL retrieves each catalog-record-field's contents, as required by IGGPLOC2.

14 IGGPSHIN places the catalog record field's contents into the user-provided work area addressed by CTGPL and increments the required work area length. If there is insufficient space in the work area, only the required work area length is changed.

15 IGGPGREC retrieves the original base catalog record, if necessary, since a horizontal extension of the base or an associated upgrade entry record may have overlaid it.

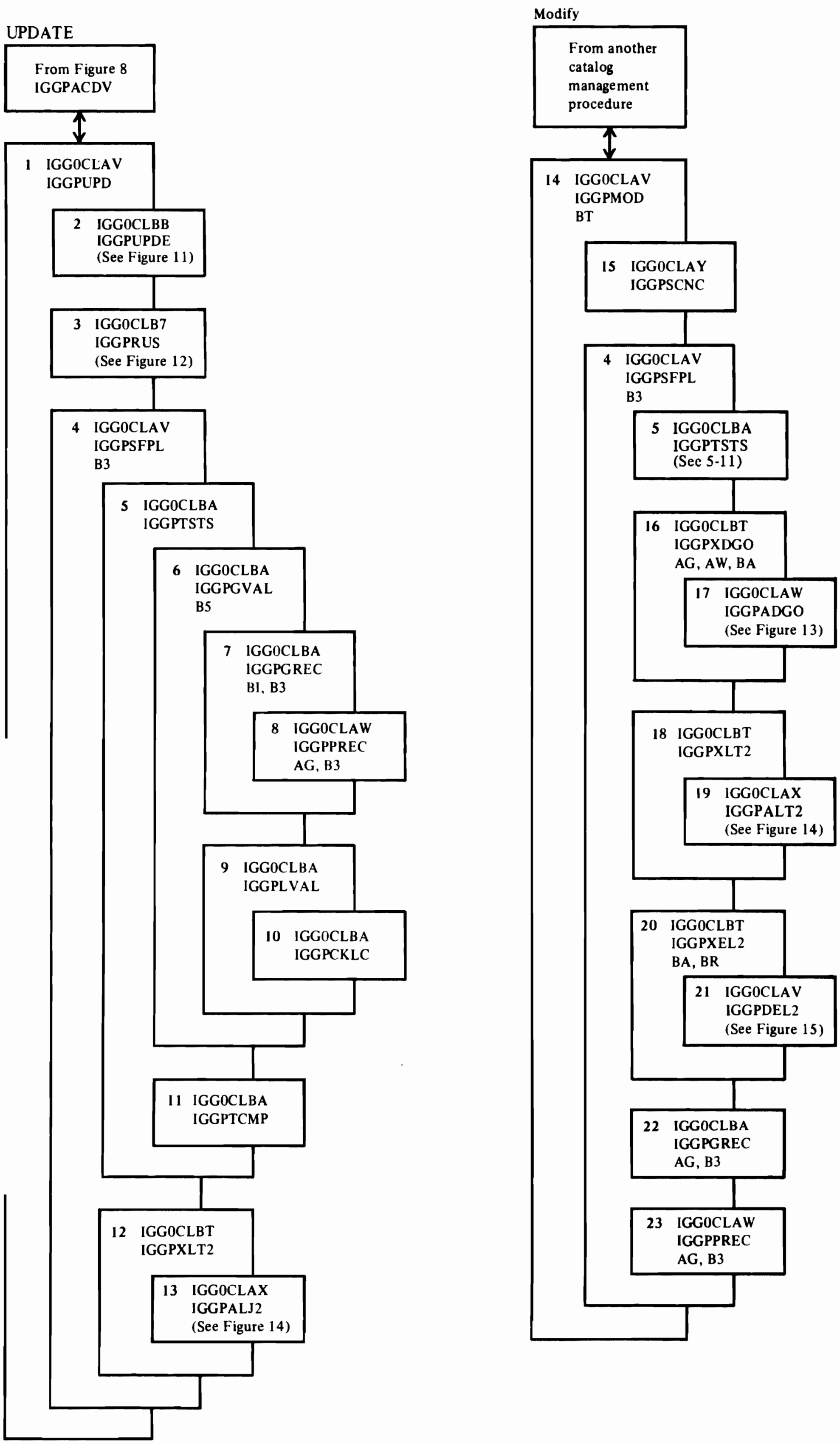


Figure 10. UPDATE/Modify Processing

Notes for Figure 10

This figure describes the processing done when a caller wants to modify a catalog record. When the caller is another part of catalog management, the caller is allowed to update the contents of a fixed-length or variable-length field, or part of a variable-length field. The caller (from within catalog management) is also allowed to delete an entire set of fields or add a new set of fields to the catalog reocrd. When the caller is an OS/VS component that issued the UPDATE command, the caller is allowed to update the contents of a fixed-length field, to request more direct-access space for a VSAM object—a data set or index—or to reset a VSAM data set.

1 IGGPUPD modifies a fixed-length catalog record field, obtain more space for a VSAM object, or calls IGGPRUS to reset a VSAM data set.

2 IGGPUPDE obtains more space for a VSAM object.

3 IGGPRUS resets a VSAM data set.

4 IGGPSFPL (**5** through **13**) processes each CTGFL addressed by the caller's CTGPL to modify all catalog record field data specified by the caller's request.

5 When the CTGFL (addressed by the CTGPL) addresses CTGFLs-for-tests, IGGPTSTS (**6** through **11**) processes each CTGFL-for-tests to identify each set of fields (group occurrence) that satisfies the test conditions.

6 IGGPGVAL retrieves one catalog-record-field's value.

If the specified field name indicates a "derived information" field, IGGOCLBS processes the field.

7 If more set-of-fields pointers (group occurrence pointers) are in an extension of the base catalog record, or

If the specified set-of-fields pointer (group occurrence pointer) contains the control interval number of an extension record,

IGGPGREC retrieves the required extension record.

BI: IGGPGET issues GET to retrieve the catalog record.

B3: IGGPSMFG makes a copy of the catalog record in case it is updated later.

8 IGGPPREC writes the contents of the buffer (a catalog record) prior to reading another record into it, if the "buffer-must-be-written" indicator is on.

AG: IGGPPUPC issues PUT-Update to rewrite an updated catalog record.

AG: IGGPPAD issues PUT-Add to insert a new catalog record into the catalog.

B3: IGGPSMF identifies the copy of the original catalog record (saved by IGGPSMFG) as an updated record.

Note: When the catalog record is completely updated, a SMF record type 63—VSAM Data Set Cataloged — is written that contains the entire new catalog record (the base and all extensions) and each part of the original catalog record that was modified (part=logical catalog record=505-byte (or less) base or extension record's contents).

9 IGGPLVAL locates the field within the catalog record.

10 IGGPCKLC verifies that the field exists (that is, the requested field is in the record or one of its sets of fields (group occurrences)).

11 IGGPTCMP compares the catalog record field's value to the caller's test data and, if the compare is OK, saves the sequence number of the catalog-record-field's set-of-fields pointer (group occurrence pointer).

12 IGGPXLT2 filters those field names (derived) which do not exist physically in the catalog. It ensures that these fields are not updated; all others are passed to IGGPALT2.

13 IGGPALT2 replaces a catalog-record-field's contents with the caller's update data.

If the caller provided CTGFLs-for-tests, each catalog record field is updated if it is:

- Identified by the CTGFL's (addressed by the CTGPL) field name, and
- Contained in a set of fields (group occurrence) that satisfies all tests associated with the CTGFL (addressed by the CTGPL). The set-of-fields pointer's (group occurrence pointer's) sequence number is available from **11**.

If the caller didn't provide CTGFLs-for-tests, each set-of-field's field identified by the CTGFL's field name is updated.

14 IGGPMOD allows an OS/VS2 catalog management procedure to update catalog record information in the following ways:

A new set of fields (group occurrence) is added to the record (IGGPXDGO processing).

A set of fields (group occurrence) is removed from the catalog record (IGGPXEL2 processing).

A fixed-length field, variable-length field, or part of a variable-length catalog record field's contents is modified (IGGPXLT2 processing).

If the specified field name indicates a "derived information" field, IGGOCLBT processes the field name and update the catalog record information implied by the field name.

15 IGGPSCNC initializes the CTGFLs with dictionary information required to find the field in the catalog record, and ensures that the CTGFLs are valid.

Notes for Figure 10 (Continued)

16 IGGPXDGO intercepts field names (derived) which do not exist physically in the catalog. All others are passed to IGGPADGO. It constructs a bit map set of fields when the first data space group is added and updates the associated data space group when data space descriptors are added. Note that derived field names exist only in the volume entry record.

AG: IGGPAOCI obtains a control interval for constructing the bit map set of fields in an extension record.

AG: IGGPPAD adds the newly constructed bit map record.

AW: IGGPPREC updates the base volume entry record which points to the bit map set of fields.

BA: IGGPGVAL retrieves the data space group associated with the space descriptor group to be added.

BA: IGGPGREC retrieves the base volume entry record so that bit map processing can be done.

BR: IGGPBMR updates the bit map to reflect the added space.

17 When the caller provides set-of-fields (group occurrence) field data, but doesn't provide CTGFLs-for-tests, IGGPADGO builds a new set-of-fields (group occurrence) with the caller's field data and adds it to the catalog record.

18 IGGPXLT2 filters those field names (derived) which do not exist physically in the catalog. It ensures that these fields are not updated; all others are passed to IGGPALT2.

19 When the caller provides header-field field data, or when the caller provides set of fields (group occurrence) field data and CTGFLs-for-tests, IGGPALT2 modifies the field's contents (as per **13** above) and makes all necessary adjustments to the catalog records.

20 IGGPXEL2 causes the bit map set of fields to be updated when a data space group is to be deleted. All set-of-field names, both derived and nonderived, are passed to IGGPDEL2.

BA: IGGPGVAL retrieves the data space group to be deleted.

BA: IGGPGREC retrieves the base volume entry record so bit map processing can be done.

BR: IGGPBMR updates the bit map to reflect the released space.

21 When the caller doesn't provide field data, IGGPDEL2 deletes catalog record sets of fields (group occurrences).

If the caller provides CTGFLs-for-tests, only those sets of fields (group occurrences) identified by IGGPTSTS (see **10**) are deleted.

If the caller didn't provide CTGFLs-for-tests, all sets of fields (group occurrences) that contain the field identified by the CTGFL's (addressed by the CTGPL) field name are deleted.

22 IGGPGREC retrieves the original base catalog record for processing the next CTGFL, since a horizontal extension record may have overlaid it.

BI: IGGPGET issues GET to retrieve the catalog record.

B3: IGGPSMFG makes a copy of the catalog record in case it is updated later.

23 IGGPPREC flushes any catalog buffers that must be written.

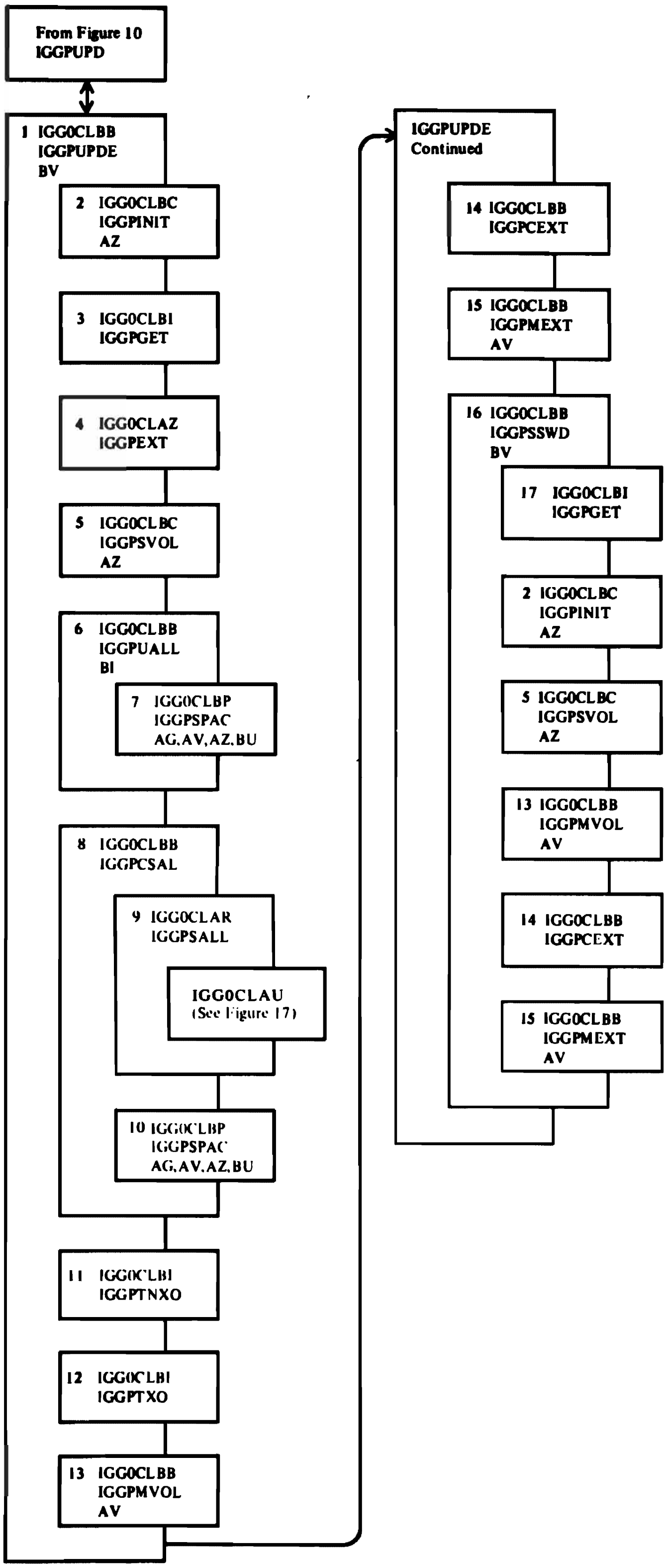


Figure 11. UPDATE/Extend Processing

Notes for Figure 11

This figure describes the processing done when a caller wants to obtain more direct-access device space for a VSAM object—a data set or an index. When the VSAM object to be extended is in a nonunique data space, the data space contains two or more VSAM objects and, possibly, enough free space to satisfy the caller's request. When the VSAM object is in a unique data space, it is the only object in the data space and there is no free space in the data space.

1 IGGPUPDE obtains more space for the VSAM object.

BV: IGGPSMFA writes SMF record type 63—VSAM Data Set Cataloged or Altered—for the data set catalog record.

BV: IGGPSMFL writes SMF record type 69—VSAM Data Space Defined, Extended, or Deleted—if additional space was obtained by DADSM, either to create a new VSAM data space or to extend an existing VSAM data space.

2 IGGPINIT initializes a CTGPL and CTGFLs and calls IGGPEXT.

AZ: IGGPEXT retrieves the AMDSB set of fields.

3 IGGPGET retrieves the base index record for key-sequenced data sets that have sequence set with data.

4 IGGPEXT retrieves the index AMDSB for key-sequenced data sets that have sequence set with data to ensure that the maximum number of extents for the index component will not be exceeded.

5 IGGPSVOL finds the volume information set of fields (group occurrence) that describes the volume that contains the VSAM object to be extended.

AZ: IGGPEXT locates the volume information set of fields.

6 When the object is in a non-shared (unique) data space, the object is allowed to reside in only one data space (16 extents maximum) per volume.

If the data space exists and contains less than 12 extents, IGGPUALL calls IGGPSPAC to obtain another extent(s) for the data space. IGGPSPAC will return a maximum of 5 extents to satisfy the request.

If the data space already contains 12 or more extents, no space is allocated to the object from the specified volume.

If the data space doesn't exist (the volume is a candidate volume for the object), IGGPUALL calls IGGPSPAC to build a data space on the volume for the object.

BI: IGGPGET issues GET to retrieve the volume catalog record.

7 IGGPSPAC calls OS/VS DADSM to obtain space for a VSAM data space (EXTEND) or to create a new VSAM data space (ALLOCATE).

AG: IGGPISCI ensures that a catalog or CRA extend will not occur while the catalog volume entry is being modified.

AG: IGGPPUPC updates the base catalog volume entry record after the new timestamp value has been set in it.

AV: IGGPMOD updates the volume catalog record fields.

AZ: IGGPEXT retrieves volume catalog record fields.

BU: IGGPF4DQ, IGGPF4RD, and IGGPF4WR update the physical volume's format-4 DSCB with new timestamp values.

8 When the object is in a shared (nonunique) data space, the object is allowed to reside in many data spaces (123 extents maximum per volume).

Note: An extent for an object in a shared data space is a contiguous amount of tracks in one of the data space's extents (obtained by OS/VS DADSM).

IGGPCSAL attempts to obtain more space for the object from any shared data space on the volume.

9 IGGPSALL calls IGGPSALS to attempt to obtain the required amount of space from the free space of one of the shared (nonunique) data spaces on the volume.

Note: All of the requested amount of space can be obtained from more than one shared (nonunique) data space. If the amount of space necessary to satisfy the request is not contiguous, IGGPSALS obtains (a maximum of five) contiguous amounts of tracks to satisfy the request.

10 When all shared (nonunique) data spaces on the volume have been examined, and none can satisfy the request, IGGPSPAC attempts to obtain another extent(s), at least large enough to satisfy the request, for any shared (nonunique) data space. If all shared data spaces have 12 or more extents, IGGPSPAC builds a new data space on the volume (if the volume contains enough contiguous free space to satisfy a data space's primary allocation requirements.) If IGGPCSAL called IGGPSPAC to obtain more space from OS/VS DADSM, IGGPCSAL calls IGGPSALL again to suballocate part of the newly obtained space to the object.

11 IPPGTNXO computes the sum of the beginning CCHHs (converted to relative track numbers) for the newly acquired extents (recoverable catalogs only).

12 IGGPTXO updates the data set directory with the new sum computed in 11.

13 IGGPMVOL updates the volume information set of fields (group occurrence) to describe the object's newly obtained space.

AV: IGGPMOD modifies the volume information set of fields (group occurrence) fixed-length fields and to modify statistical information in the AMDSB.

Notes for Figure 11 (Continued)

14 IGGPCEXT builds extent descriptors to insert in the volume information set of fields. The extent descriptors describe each contiguous amount of newly allocated space. IGGPCEXT computes the RBAs and CCHH values from the information returned by IGGPSALL:

CCHH NN Desc#

where:

CCHH is the starting cylinder and track number of the extent,

NN is the number of tracks in the extent, and

Desc# is the data space descriptor's sequence number.

15 IGGPMEXT inserts the extent descriptors into the volume-information set of fields.

AV: IGGPMOD adds the extent descriptor(s) to the volume information set of fields (group occurrence).

16 When the space is obtained for a sequence-set-with-data object (data set or key range), IGGPSSWD allocates part of the newly obtained space to the object's sequence set and modifies the object's index catalog records to reflect this.

IGGPGET retrieves the index catalog record. The index entry is then processed in a manner similar to the data entry except that space has already been acquired (hence, neither IGGPUALL nor IGGPCSAL is called).

BV: IGGPSMFA writes SMF record type 63—VSAM Data Set Cataloged or Altered—for the index catalog record.

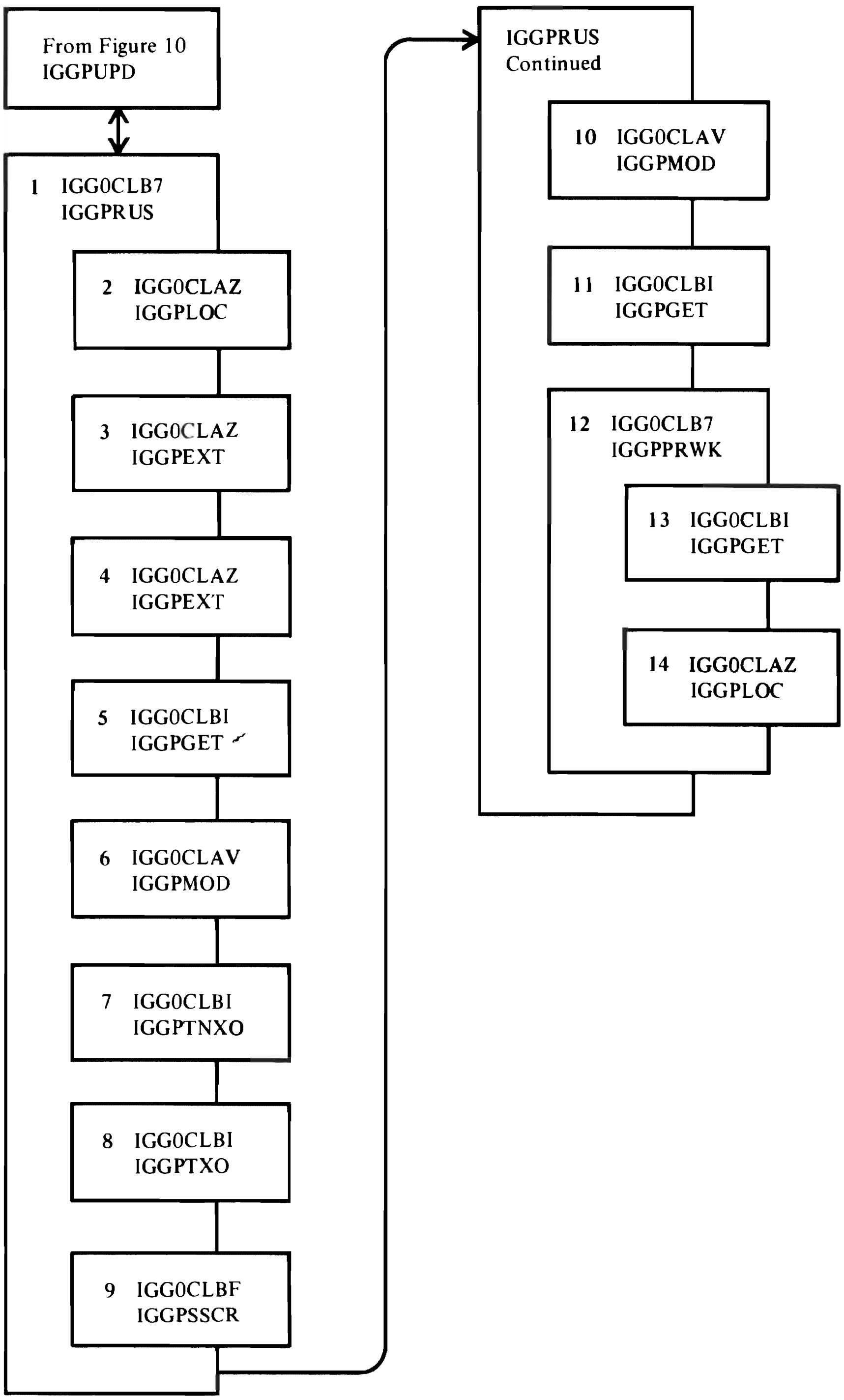


Figure 12. Reusable Data Set Processing

Notes for Figure 12

This figure describes the processing done when a caller wants to reset (reuse) a VSAM data set. The reusable attribute may be applied to a VSAM key-sequenced, entry-sequenced, or relative record data set and to an alternate index. The attribute allows the data set to have its high-used RBA set to 0 at OPEN time. Reusable data sets may be multi-volumed and they must be suballocated (nonunique). Also, they cannot have key ranges and they are restricted to a maximum of 16 physical extents per volume. A reusable base cluster may not have alternate indexes; however, reusable alternate indexes can be associated with a nonreusable base cluster.

1 IGGPRUS checks the data set catalog record to verify that it is a type D record and that the data set has the correct attributes for resetting. Appropriate error codes are returned to the caller if data set does not have the proper attributes.

2 If the data set is not reusable (but is empty), IGGPRUS calls IGGPLOC before returning to caller. No error codes are returned.

3 If the data set can be reset, IGGPEXT retrieves the AMDSB set of fields. If the data set attributes in the AMDSB indicate a key-sequenced data set, the user's work area (CTGWKA) is checked to ensure that it contains the control interval number of the index record in the catalog. If not, return to caller with the appropriate error code.

4 IGGPEXT retrieves each volume information set of fields pertaining to the data set.

5 IGGPGET retrieves the corresponding volume records from the catalog.

6 Each volume information set of fields is updated so as to retain only primary extents. If no primary extents remain after updating, the volume becomes a candidate volume. IGGPMOD returns the updated volume information set of fields to the catalog.

7 If a recoverable catalog is involved, IGGPTNXO calculates for each volume processed the sum of the relative track addressed of extents remaining in the reset data set.

8 IGGPIXO uses the value from step 7 to update the Data Set Directory set of fields for each volume processed.

9 If the resetting of the data set results in extents being freed on the volume being processed, IGGPSSCR updates the Space Map set of fields.

10 IGGPMOD updates the catalog with the AMDSB set of fields and the base data set catalog record.

11 If a key-sequenced data set is being reset, IGGPGET retrieves the index catalog record and resetting of the index section of the data set proceeds as described above for the data component.

12 If a key-sequenced data set has been reset, IGGPFRWK retrieves the data set catalog record required by IGGPLOC.

13 IGGPLOC processes the user's LOCATE request.

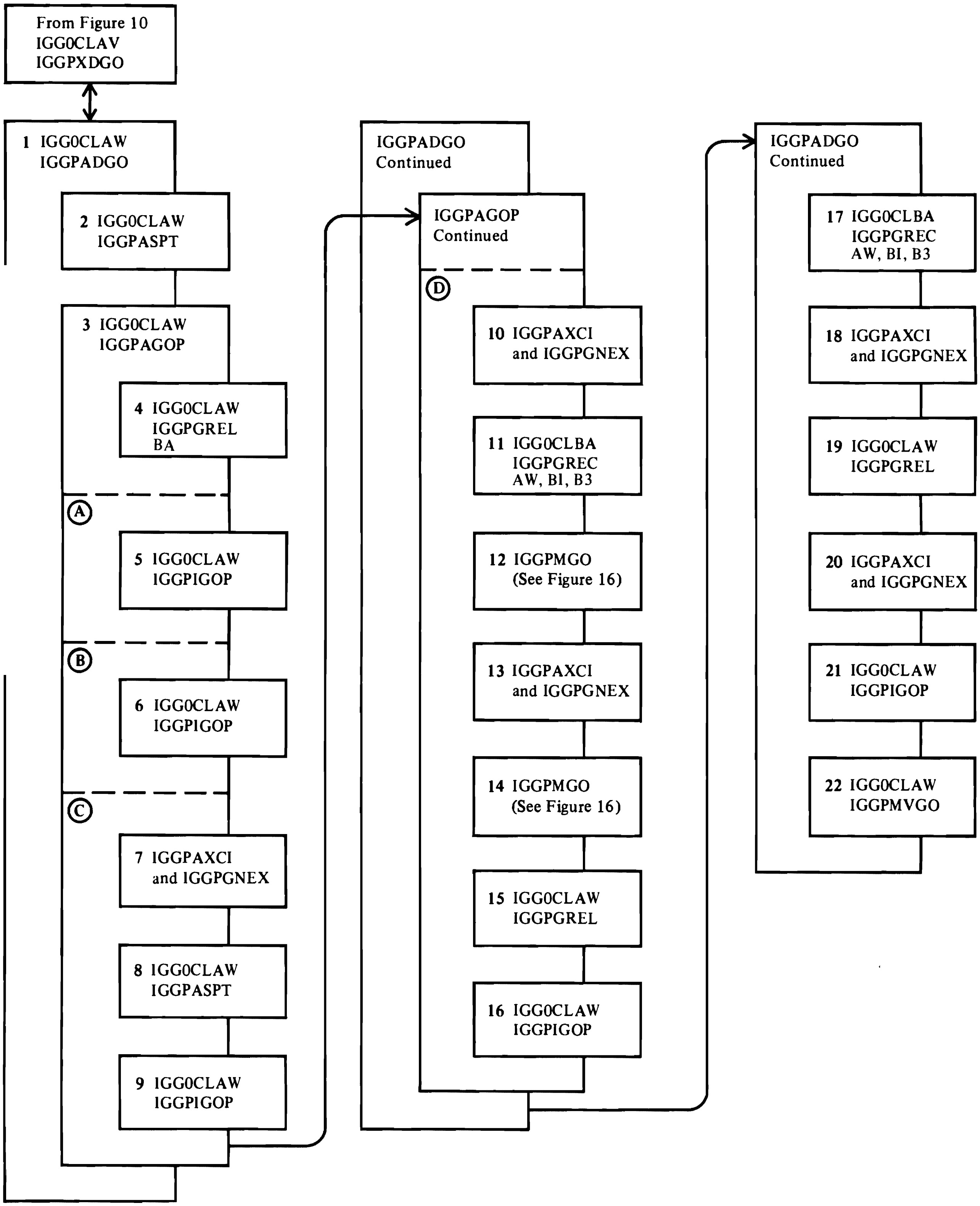


Figure 13. Insert a New Set of Fields (IGGPADGO Processing)

Notes for Figure 13

This figure describes the processing done when a catalog management routine needs to insert a new set of fields into a catalog record to satisfy the caller's request.

1 IGGPADGO inserts a new set of fields (group occurrence) into the object's catalog record and adjusts other catalog records as required. In the notes for this figure, the hypothetical set of fields to be added is called ADDSET.

Note: If ADDSET can fit in the base catalog record, its set-of-fields pointer (group occurrence pointer) in the base catalog record contains ADDSET's displacement from the beginning of the base catalog record's sets of fields (group occurrences).

Otherwise, ADDSET is put in an extension record and two set-of-fields pointers (group occurrence pointers) are used to locate it. Its set-of-fields pointer in the base catalog record contains the extension record's control interval number. ADDSET's set-of-fields pointer in the extension record contains its displacement from the beginning of the extension records sets of fields.

2 If the base catalog record doesn't contain an "available-space pointer" (a pointer to an extension record that contains free space), IGGPASPT builds an "available-space pointer" (a set-of-fields pointer with sequence number = 0 and type (group) code = 0) and calls IGGPIGOP to insert it into the base catalog record.

3 IGGPAGOP ensures that there is a set-of-fields pointer (group occurrence pointer) in the base catalog record for ADDSET and determines where the set of fields can be inserted. If the base catalog record (or one of its extensions) contains a deleted set-of-fields pointer (with the same type code as ADDSET), IGGPAGOP activates it and assigns it to ADDSET. If not, IGGPAGOP builds a set-of-fields pointer and inserts it into the base catalog record (or one of its extensions).

4 IGGPGREL identifies the set-of-fields pointer (group occurrence pointer) in the base catalog record (or one of its extensions) that has the same type code as ADDSET and either:

- Is marked as deleted, or
- Has the highest sequence number with ADDSET's group code.

BA: IGGPGREC retrieves any horizontal extensions of the base entry record.

IGGPAGOP then ensures that the base catalog record contains a set-of-fields pointer (group occurrence pointer) that will point to ADDSET:

A If IGGPGREL found a set-of-fields pointer (group occurrence pointer) marked deleted which points to an extension record:

5 IGGPIGOP activates the set-of-fields pointer. IGGPAGOP is finished; ADDSET is to be inserted into the extension record pointed to by the set-of-fields pointer.

B If ADDSET and its set-of-fields pointer (group occurrence pointer) can fit in the base catalog record:

6 IGGPIGOP builds a set-of-fields pointer for ADDSET and inserts it into the base catalog record following the set-of-fields pointer identified by IGGPGREL (see **4**). IGGPAGOP is finished; ADDSET is to be inserted into the base catalog record.

C If the base catalog record cannot contain a new set-of-fields pointer (group occurrence pointer), even if all sets of fields are moved out of the base catalog record:

7 IGGPAXCI and IGGPGNEX obtain an extension record for the base catalog record.

If the record area in which the extension record is to be built already contains a record to be written, IGGPPREC flushes the record.

8 IGGPASPT builds an available space pointer and inserts it into the base catalog record's extension.

9 IGGPIGOP builds a set-of-fields pointer for ADDSET and inserts it into the base catalog record's newly obtained extension record. IGGPAGOP is finished; ADDSET is to be inserted into the base catalog record's newly obtained extension.

D If the base catalog record can contain a new set-of-fields pointer (group occurrence pointer) by moving the sets of fields (group occurrences) in the base catalog record into an extension record, move the set of fields out of the base catalog record into an extension record and adjust the base catalog record as necessary:

10 If the base catalog record's available-space pointer doesn't point to an extension record, IGGPAXCI and IGGPGNEX obtain an extension record to contain the sets of fields.

11 If the base catalog record's available-space pointer points to an extension record, IGGPGREC retrieves the extension record.

AW: IGGPPREC writes any record in the record area that must be flushed before the extension record can be retrieved.

BI: IGGPGET retrieves the extension record.

B3: IGGPSMFG ensures that a copy of the original extension record exists.

12 IGGPMGO moves *all* sets of fields (except type codes 1 or 2) into the extension record from the base catalog record. IGGPMGO adjusts each set-of-fields pointer in the base catalog record to point to the extension record containing its set of fields.

13 If the extension record obtained by **11** isn't able to contain all of the base record's sets of fields, IGGPAXCI and IGGPGNEX obtain another extension record to contain the remaining sets of fields. IGGPAGOP updates the base catalog record's available-space pointer to point to the newly obtained extension record.

14 IGGPMGO moves the rest of the base catalog record's sets of fields into the extension record (obtained in **13**).

Notes for Figure 13 (Continued)

15 IGGPGREL re-identifies the set-of-fields pointer (group occurrence pointer) in the base catalog record for ADDSETs use.

16 IGGPIGOP activates or builds a new set-of-fields pointer for ADDSET. IGGPAGOP is finished; the record into which ADDSET is to be inserted is the last extension record obtained by 11 or 13.

If ADDSET is not to be added to the base catalog record and if the (deleted, now activated) set-of-fields pointer (see **3**) points to an extension extension record:

17 IGGPGREC retrieves the extension record into which ADDSET is to be inserted. (See Figure 10, **7** through **8** for details.)

18 If ADDSET and its set-of-fields pointer cannot fit in the extension record, IGGPAXCI and IGGPGNEX obtain another extension reocrd.

19 IGGPGREL determines where (in the extension record) ADDSET's set-of-fields pointer (group occurrence pointer) should be be inserted.

If ADDSET is not to be added to the base catalog record, and if ADDSET's set-of-fields pointer (in the base catalog record) doesn't point to an extension record:

20 IGGPAXCI and IGGPGNEX obtain an extension record to contain ADDSET.

21 IGGPIGOP builds a new set-of-fields pointer (group occurrence pointer) for ADDSET and inserts it into the extension record that will contain ADDSET.

22 IGGPMVGO moves ADDSET into the record that is to contain it.

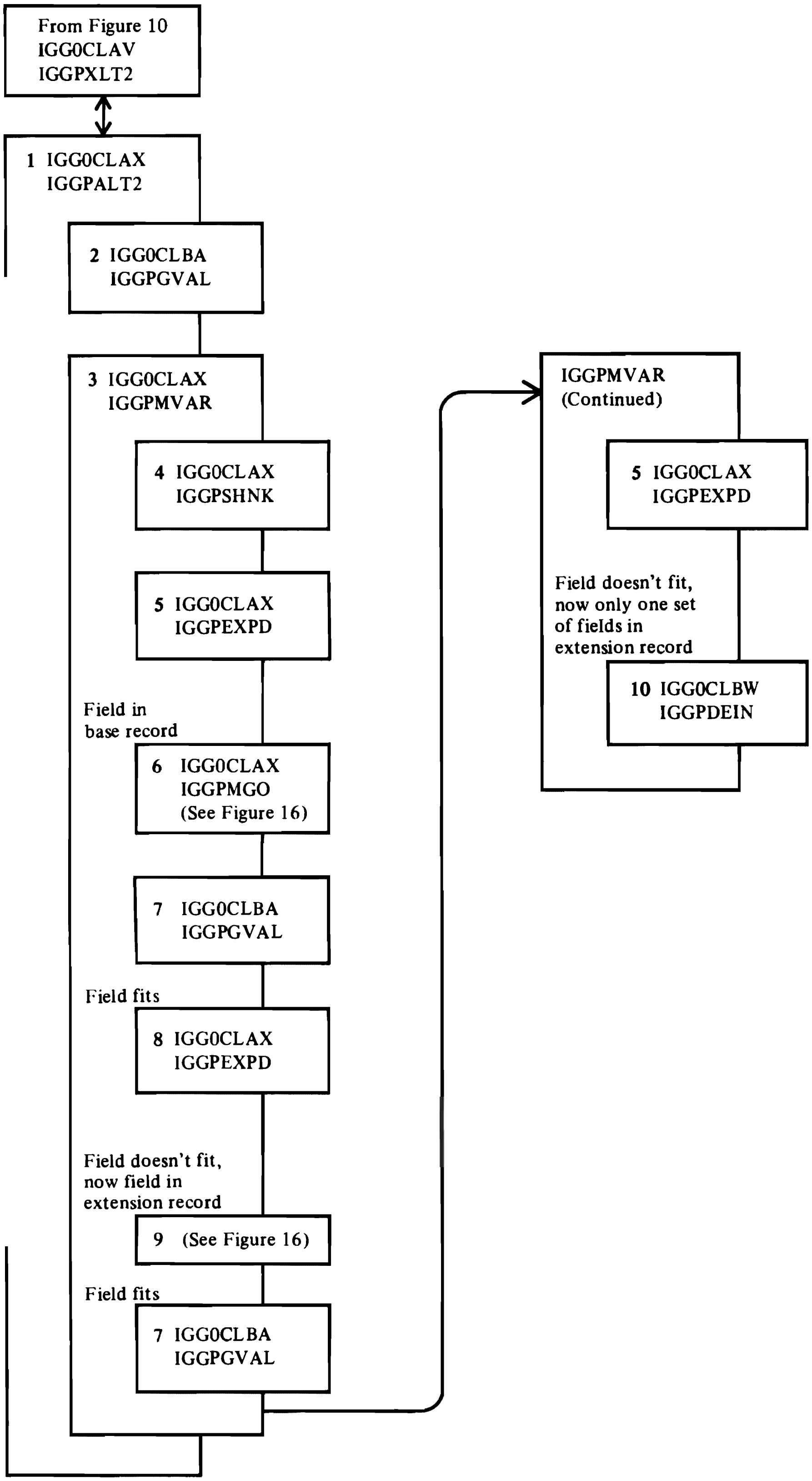


Figure 14. Modify Field Data (IGGPALT2 Processing)

Notes for Figure 14

This figure describes the processing done when a caller needs to update a field in a catalog record. When the caller is an OS/VS component that issued the UPDATE, the caller is only allowed to replace the contents of a fixed-length field with updating data. When the caller is a catalog management routine, the caller is allowed to modify variable-length fields, which might require additional processing to accomodate a larger field.

1 IGGPALT2 modifies the contents of a catalog record. It repeats the following sequence to process each field name identified by a combination name.

2 IGGPGVAL retrieves the field to be modified. (See Figure 10, **5** through **9** for for details.)

3 When the field is fixed-length, IGGPALT2 replaces the field's contents with the caller's update data.

When the field is variable-length with length change, IGGPMVAR replaces the field's contents with the caller's update data and adjusts other catalog records as required.

Note: IGGPMVAR can only be used with IGGPMOD callers, since IGGPUPD callers do not modify the contents of variable-length fields.

4 When the variable-length field's new length is less than or equal to its old length, IGGPSHNK:

1. Adjusts the rest of the record so that all the record's free space is contiguous, and increases the amount of free space.
2. Adjusts other set-of-fields pointers (group occurrence pointers) to reflect displacement changes resulting from the free space adjustment.
3. Replaces the catalog record field's contents with the caller's update data.

5 When the variable-length field's new length is greater than its old length, and when the entire new field's contents can be contained in the catalog record, IGGPEXPD:

1. Adjusts the rest of the record so that the larger field is inserted, and decreases the amount of free space.
2. Adjusts other set-of-fields pointers (group occurrence pointers) to reflect displacement changes resulting from the insertion.
3. Replaces the catalog record field's contents with the caller's update data.

When the entire field's new contents cannot be contained in the catalog record, and that record is the base catalog record:

6 IGGPMBGO moves the field's set of fields (group occurrence) into an extension record.

7 IGGPGVAL locates the field in the extension reocrd.

8 If the entire field's new contents can be contained in the extension record, IGGPEXPD updates the field's contents as described in **5**.

When the entire field's new contents cannot be contained in the catalog record; when the record is an extension record, and when the extension record contains two or more sets of fields (group occurrences):

9 Sets of fields (group occurrences) are moved out of the extension record into another extension record (as described in Figure 16) until:

- The entire field's new contents can be contained in its set-of-fields' extension record, or
- The field's set of fields (group occurrence) is the only set of fields in an extension record.

When the entire field's new contents cannot be contained in the catalog record; when the record is an extension record; and when the extension record contains only the field's (to be updated) set of fields (group occurrence):

10 IGGPDEIN updates the field's contents and adjusts other catalog records as required.

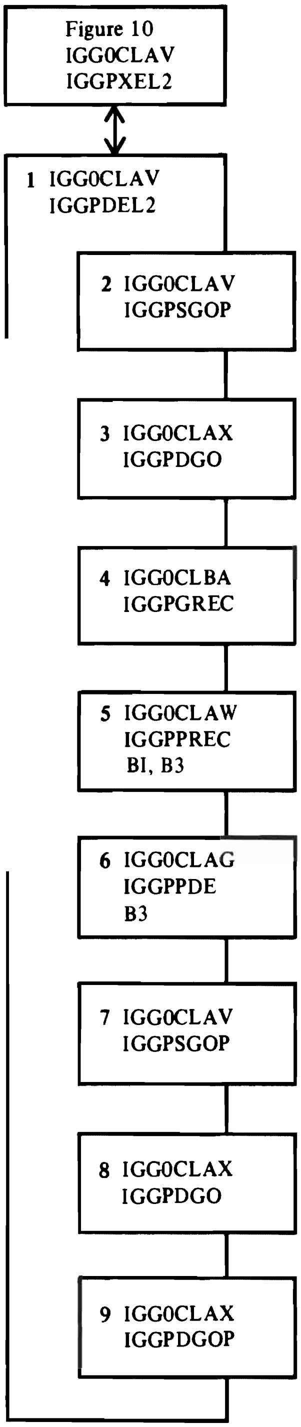


Figure 15. Remove a Set of Fields (IGGPDEL2 Processing)

Notes for Figure 15

This figure describes the processing done when a catalog management routine needs to delete a set of fields from a catalog record to satisfy the caller's request.

1 IGGPDEL2 removes a set of fields (group occurrence) and adjusts other catalog records as required. When IGGPXEL2 calls IGGPDEL2, IGGPSFPL has determined the sequence number of the set-of-fields pointer (group occurrence pointer) to be deleted. In this figure, the set of fields to be deleted is called DELSET.

2 IGGPSGOP locates DELSET's set-of-fields pointer (group occurrence pointer) in the base catalog record or one of its horizontal extensions.

When DELSET is in the base catalog record or one of its horizontal extensions:

3 IGGPDGO deletes DELSET. IGGPDEL2 marks its set-of-fields pointer (group occurrence pointer) as deleted and zeros its displacement value. DELSET is now deleted; IGGPDEL2 is finished and returns to IGGPSFPL.

When DELSET is in an extension record:

4 IGGPGREC retrieves the extension record and as many of its extensions as required to delete DELSET. (See Figure 10, **6** through **7** for details.)

When DELSET is the only set of fields in the extension record:

IGGPDEL2 marks DELSET's set-of-fields pointer (group occurrence pointer) in the base catalog record as deleted and zeros its pointer to the extension record.

5 IGGPPREC updates the object's base catalog record.

6 IGGPPDE issues PUT-Update to rewrite the extension as a free catalog control interval.

B3: IGGPSMF identifies the copy of the original catalog record (saved by IGGPSMFG) as an updated record.

DELSET is now deleted; IGGPDEL2 processes each additional vertical extension that contains part of the set of fields (using steps **4** through **6**) and, when DELSET is completely deleted, returns to IGGPSFPL.

When DELSET is not the only set of fields (group occurrence) in the extension record:

7 IGGPSGOP finds DELSET's set-of-fields pointer (group occurrence pointer) in the extension record.

8 IGGPDGO deletes DELSET in the extension record.

9 IGGPDGOP deletes DELSET's set-of-fields pointer (group occurrence pointer) in the extension record.

IGGPDEL2 marks DELSET's set-of-fields pointer (group occurrence pointer) in the base catalog record as deleted, and zeros its pointer to the extension record. DELSET is now deleted; IGGPDEL2 is finished and returns to IGGPSFPL.

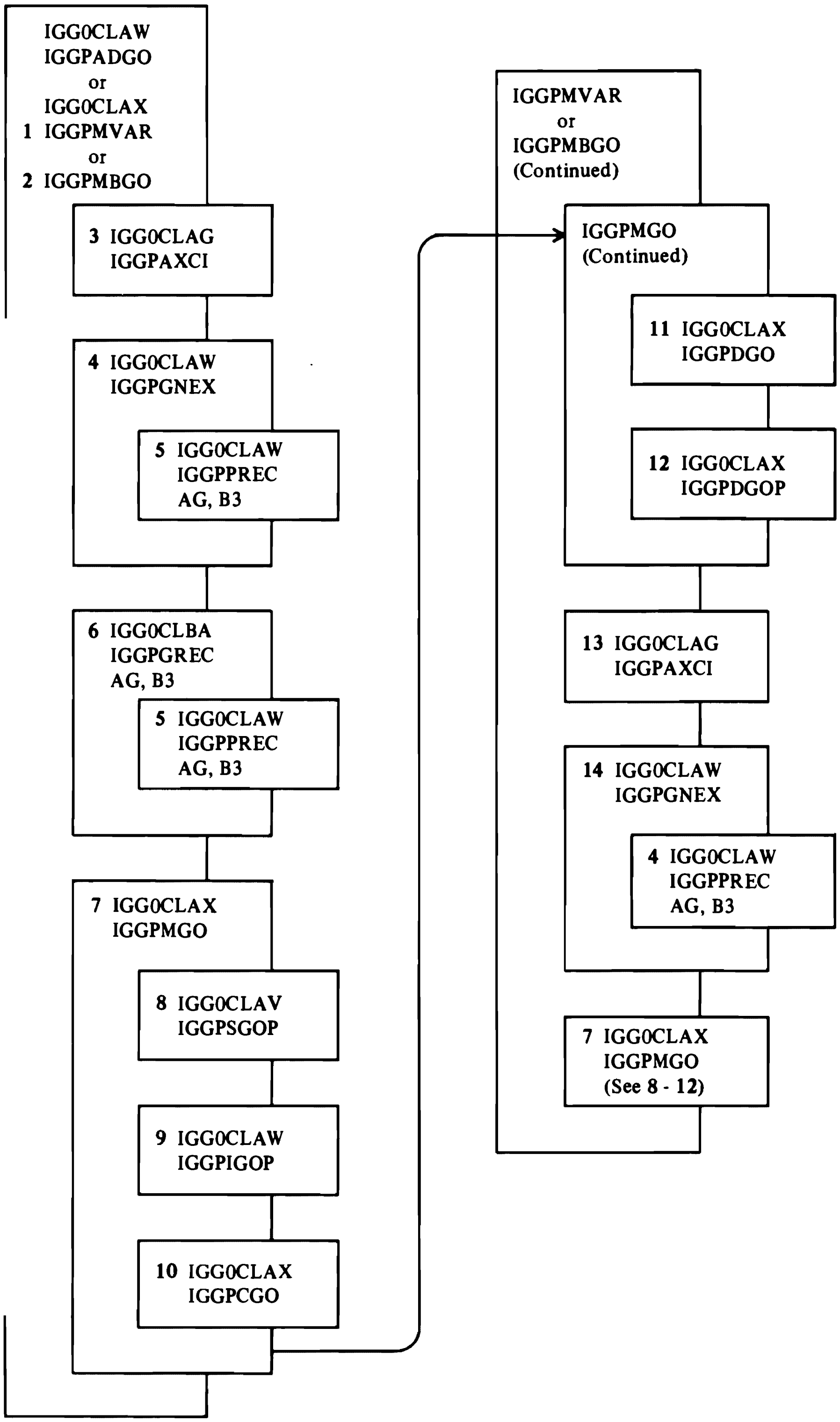


Figure 16. Move a Set of Fields from a Catalog Record into its Extension

Notes for Figure 16

This figure describes the sequence required to move a set of fields (group occurrence) from one record (OLDREC) to another (NEWREC). The acronyms "OLDREC" and "NEWREC" are introduced to help you understand this process; these acronyms do not appear in the catalog management code as symbolic names or comments.

1 IGGPMVAR executes this sequence to move each set of fields (group occurrence), except the one that contains the expanded variable-length field, from one extension record to another. IGGPMVAR moves one set of fields at a time, the record's last set of fields, and returns to its caller.

2 IGGPMBGO executes this sequence to move the expanding set of fields (group occurrence) in the base catalog record into an extension record.

3 When the base catalog record's available-space pointer (identified by type (group) code = 0 and sequence number = 0) is zero, IGGPAXCI obtains a free control interval.

4 IGGPGNEX initializes the free control interval as an extension record (NEWREC). IGGPGNEX then updates the available-space pointer to contain NEWREC's control interval number.

5 If the "buffer must be written" indicator is on, IGGPREC writes the contents of the buffer (a catalog record) prior to reading another record into it.

AG: IGGPPUPC rewrites an updated catalog record.

AG: IGGPPAD writes a new (extension) catalog record.

B3: IGGPSMF identifies the copy of the original catalog record (saved by IGGPSMFG) as an updated record.

6 When the base catalog record's available-space pointer points to an extension record with free space, IGGPGREC retrieves the extension record (NEWREC).

AG: IGGPGET issues GET to retrieve the catalog record.

B3: IGGPSMFG makes a copy of the catalog record in case it is updated later.

7 When there is enough free space in NEWREC to contain OLDREC's set of fields (group occurrence), IGGPMGO moves the set of fields (group occurrence) from OLDREC to NEWREC.

8 IGGPSGOP searches NEWREC to locate the position of the new set-of-fields' pointer (group occurrence pointer).

9 IGGPIGOP inserts a new set-of-fields pointer (group occurrence pointer) in NEWREC to contain the displacement of the new set of fields.

10 IGGPCGO copies the contents of the set of fields (group occurrence) into NEWREC and reduces the amount of NEWREC's free space.

11 IGGPDGO deletes the set of fields (group occurrence) in OLDREC and increases the amount of OLDREC's free space.

12 If OLDREC is not the base catalog record, IGGPDGOP deletes the set-of-fields pointer (group occurrence pointer) in OLDREC.

13 If the caller in step 1 or 2 determines (at **3** or **6**) that NEWREC cannot contain the set of fields (group occurrence), another extension record is built. IGGPAXCI obtains a free control interval and IGGPGNEX initializes it as an extension record (NEWREC). The base catalog record's available-space pointer is updated.

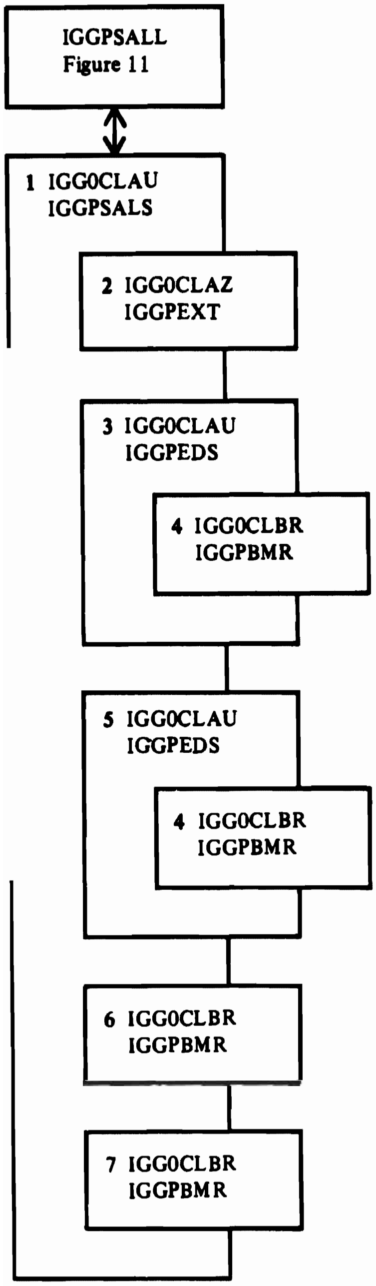


Figure 17. Allocating Part of a Data Space's Space (IGGPSALS Processing)

Notes for Figure 17

This figure describes the processing done when a catalog management routine needs to obtain an amount of direct-access device space from a nonunique data space (one that contains part or all of two or more VSAM objects) to satisfy the caller's request.

1 IGGPSALS assigns tracks to a VSAM object that can reside in a nonunique data space (a data space that can contain more than one VSAM object).

2 IGGPEXT retrieves data space group sets of fields from the volume catalog record. Each data space group set of fields contains one or more (up to 16) extent descriptors that describe the data space's extents. IGGPSALS builds a table that contains extent descriptors in the form:

S#TTNN

where

S# is the sequence number of the extent descriptor in its data space group set of fields.

TT is the extent's starting track number, and is converted by another routine to CCHH for a seek address.

NN is the number of tracks in the extent.

When the extent table is full, or when there are no more data space group sets of fields to process, IGGPSALS calls IGGPEDS to process each entry in the table.

3 IGGPEDS scans the data space extent table to find the entry with the smallest TT value. This entry is processed, then IGGPEDS scans the table again to find the entry with the next higher TT value. All entries in the table are found, then processed, from the lowest TT value to the highest.

4 IGGPBMR scans the space map set of fields, starting at bit position TT and ending at bit position TT+NN-1, attempting to find a contiguous amount of unallocated tracks large enough to satisfy the minimum allocation unit for the request (usually a control interval).

IGGPBMR returns to IGGPEDS with either a "no extent found" indicator or a TTNN value. IGGPEDS analyzes the TTNN value to determine:

1. If the extent exactly satisfies the caller's allocation request, no further extent table processing is done.
2. If the extent is larger than the caller's allocation request, but is smaller than any previously obtained extent, the (smaller) extent's TTNN and its data space's sequence number is saved. Processing the data space extent table continues.
3. If the extent is smaller than the caller's allocation request, its TTNN and data space sequence number is put in the "small extent table" if:
   - There are fewer than the maximum number of entries in the small extent table (five, or a caller-specified maximum less than five), or
   - The extent's NN value is larger than the smallest NN value in the small extent table. Processing the data space extent table continues.

5 When the data space extent table is partially full, IGGPEDS processes each entry as described in **3** and **4**.

6 IGGPBMR adjusts the bits in the space map set of fields if the caller's allocation request is satisfied with one extent.

7 If the caller's allocation request is satisfied with more than one extent, all entries in the small extent table are sorted on decreasing NN value, so that space is allocated from the least number of extents. IGGPBMR adjusts the bits in the space map set of fields for each extent required to exactly satisfy the caller's request. Each bit in the space map set of fields identifies a track on the volume as either allocated to a VSAM object or unallocated.

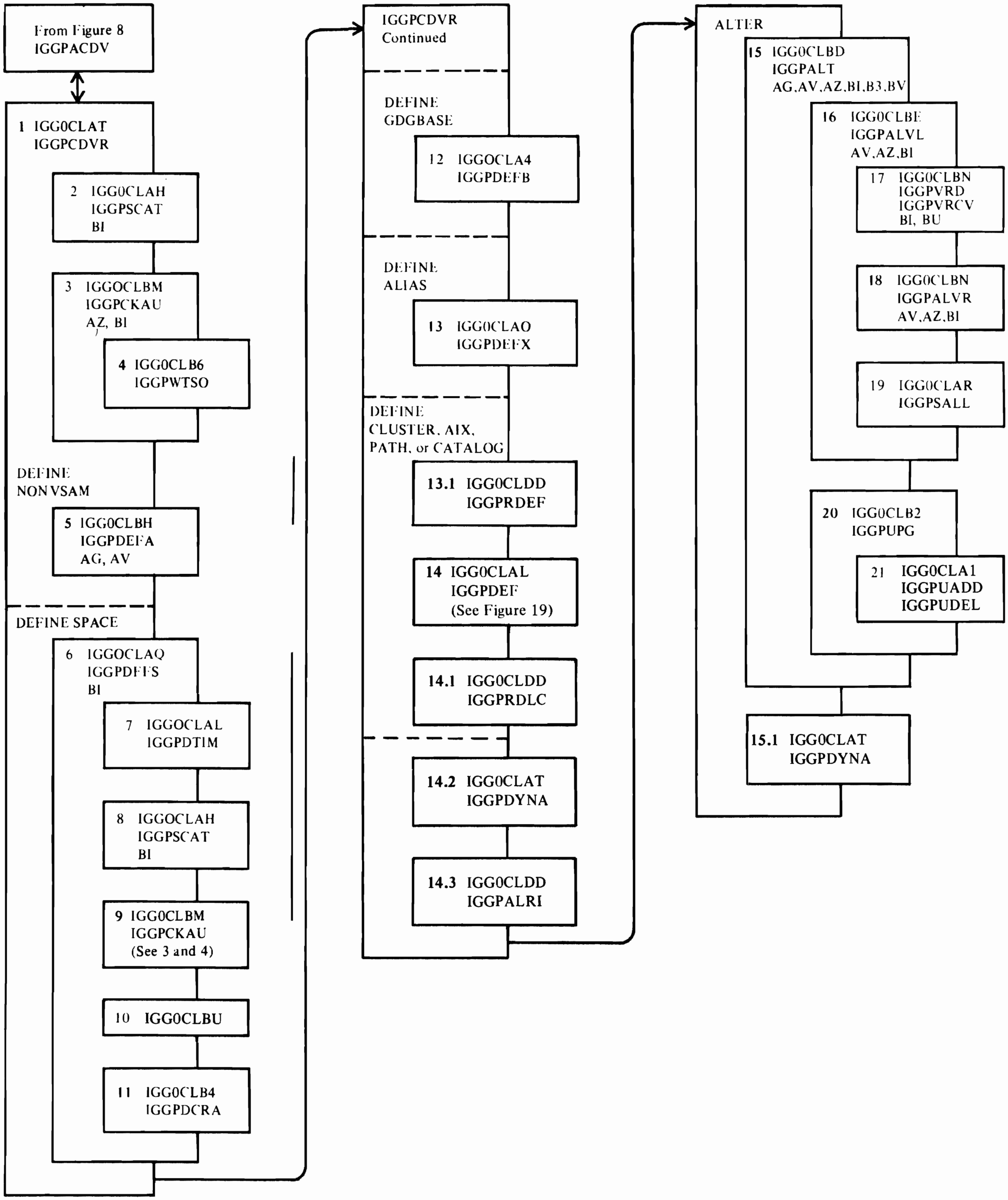


Figure 18 (Part 1 of 2). Catalog Management Services Processing

Notes for Figure 18

This figure describes the overall processing of Catalog Management Services—that group of modules that allow the user's program to add, modify, delete, and list catalog information by using Access Method Services commands. The Access Method Services commands and their parameters are described in *OS/VS2 Access Method Services*. A detailed description of the flow of control for DEFINE CATALOG and DEFINE CLUSTER processing is on the following compendium. Method of Operation Diagram EA1, Catalog Management Services Table of Contents, lists the Method of Operation Diagrams that describe Catalog Management Services functions.

1 IGGPCDVR is the Catalog Management Services Common Processing procedure.

2 IGGPSCAT searches the catalog for a duplicate name (if the request is DEFINE), or to retrieve the object's catalog record (if the request is ALTER, DELETE, or LISTCAT). If the request is DEFINE SPACE, step 3 is not performed.

3 IGGPCKAU verifies the caller's authorization to perform the request. If the request is LISTCAT, step 4 is bypassed.

AZ: IGGPEXT locates the password information required by IGGPCKAU.

BI: IGGPGET issues GET to retrieve the catalog's cluster catalog record (control interval number 2) to determine if the catalog is password protected. If the catalog is not password protected, all security verification processing is bypassed.

4 When the user is on a TSO terminal, IGGPWTSO issues requests to the TSO terminal for the required password if the password is not supplied with the input parameters (part of the Access Method Services job syntax language).

5 When the caller's request is DEFINE NONVSAM, IGGPDEFA processes it.

AG: IGGPAOCI obtains a free control interval to contain the nonVSAM catalog record.

AV: IGGPMOD inserts information into the newly created nonVSAM catalog record.

BV: IGGPSMFA writes SMF record type 63—VSAM Data Set Cataloged.

6 When the caller's request is DEFINE SPACE, IGGPDEFS processes it.

AG: IGGPPAD issues PUT-Add to add records to the catalog as required.

BI: IGGPGET issues GET to retrieve the volume catalog record.

7 IGGPDTIM obtains time-of-day data.

8 IGGPSCAT retrieves the the catalog record identified by the CTGPL (the VSAM object's base catalog record). Extensions to the base catalog record are retrieved as they are needed.

BI: IGGPGET issues GET to retrieve catalog records.

9 IGGPCKAU verifies the caller's authorization to create a VSAM data space.

AZ: IGGPEXT locates the password information required by IGGPCKAU.

BI: IGGPGET issues GET to retrieve the volume-owner's catalog record that contains the password set of fields (group occurrence). (A OS/VS2 master or user's catalog is identified as the volume owner.)

When the user is on a TSO terminal, IGGPWTSO issues requests to the TSO terminal for the required password.

10 IGG0CLBU processes the format-4 DSCB, when the DEFINE SPACE creates the volume's first data space:

IGGPF4RD reads the format-4 DSCB.

IGGPF4DQ dequeues the volume identified by the format-4 DSCB.

IGGPF4WR writes (or updates, if it exists) the format-4 DSCB.

11 IGGPDCRA defines a catalog recovery area on the volume when the volume's first data space is created and the owning catalog is recoverable.

12 IGGPDEFB builds the GDG base catalog record and inserts it into the catalog.

13 IGGPDEFX builds the alias name catalog record. If the alias is the object's first alias, IGGPDEFX adds an association set of fields to the object's catalog record to point to the alias name catalog record. If the object has other aliases, IGGPDEFX modifies the previous alias name catalog record to point to the currently-built alias name catalog record.

13.1 IGGPRDEF issues the RACDEF SVC (133) for all user-supplied entity names when the data set is to be protected by RACF.

14 When the caller's request is to create a VSAM cluster, path, AIX, or a user's catalog, IGGPDEF processes it.

14.1 IGGPRDLC issues the RACDEF SVC (133) to delete the RACF indicator, if an error occurs in IGGPDEF.

14.2 IGGPDYNA is invoked if the RACF indicator is to be altered in a recoverable catalog and no DD statement for the CRA has been supplied by the caller. This routine invokes dynamic allocation (SVC 99) to allocate the CRA volume.

14.3 IGGPALRI is called to set or clear the RACF indicators.

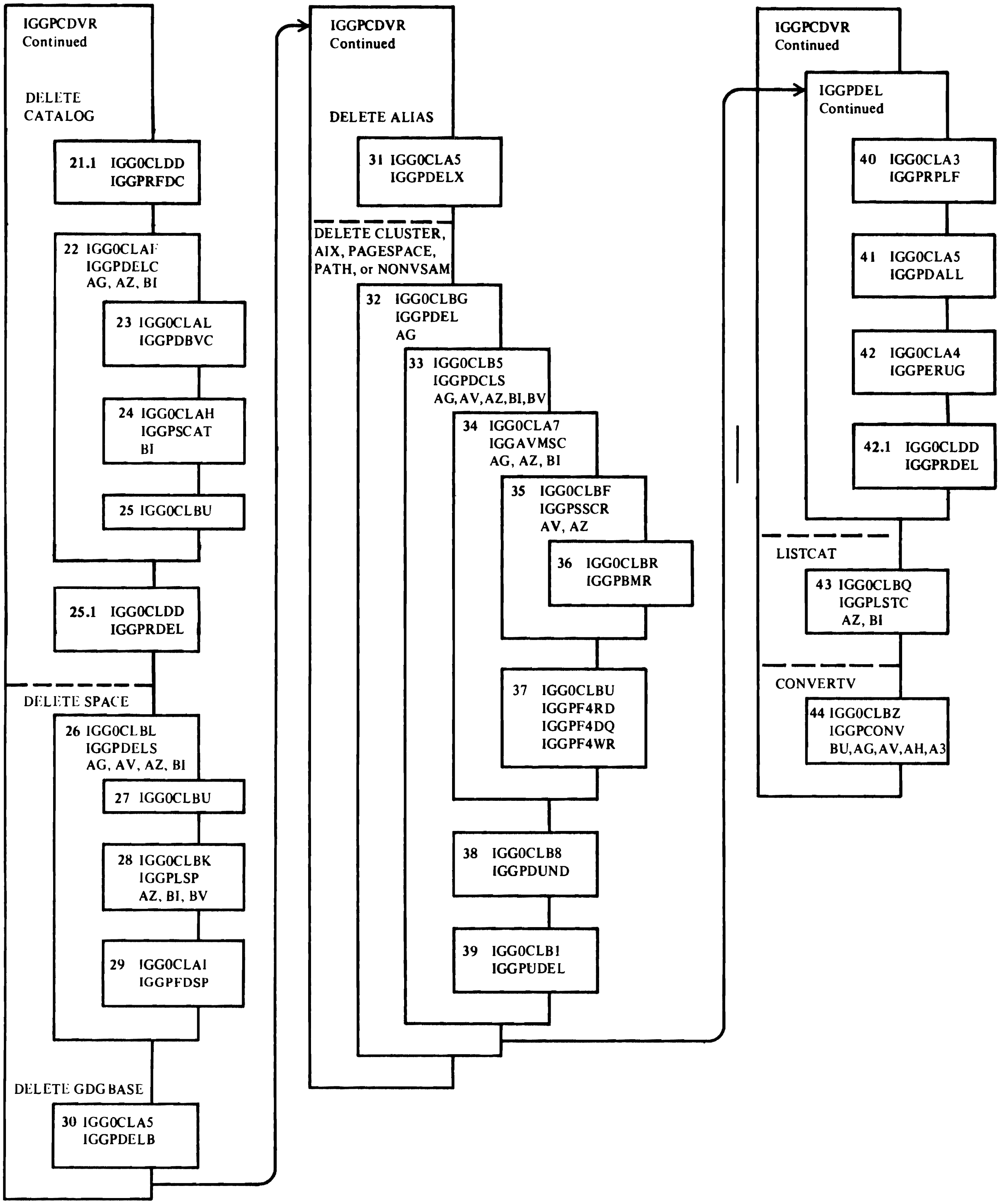


Figure 18 (Part 2 of 2). Catalog Management Services Processing

Notes for Figure 18 (Continued)

15 When the caller's request is ALTER, IGGPALT processes it.

AG: IGGPPUPC issues PUT-Update to rewrite a catalog record.

AG: IGGPPAD issues PUT-Add to insert a catalog record into the catalog.

AG: IGGPPDE issues PUT-Update to rewrite the record as a free catalog record.

AV: IGGPMOD modifies the contents of catalog record fields.

AZ: IGGPEXT locates catalog record fields.

BI: IGGPGET issues GET to retrieve a catalog record.

BV: IGGPSMFA writes SMF record type 63—VSAM Data Set Cataloged.

15.1 IGGPDYNA deallocates the CRA volume.

16 IGGPALVL modifies volume catalog record information.

AV: IGGPMOD modifies the contents of catalog record fields.

AZ: IGGPEXT locates catalog record fields.

BI: IGGPGET issues GET to retrieve the volume catalog record.

17 IGGPVRD mounts the required volume(s) and IGGPVRCV removes VSAM ownership from the volume(s).

BI: IGGPGET retrieves the volume catalog record.

BU: IGGPF4RD reads the format-4 DSCB.

BU: IGGPF4DQ dequeues the volume identified by the format-4 DSCB.

BU: IGGPF4WR writes (updates) the format-4 DSCB.

18 IGGPALVR updates the volume catalog record.

AG: IGGPPUPC rewrites the updated volume catalog record.

AV: IGGPMOD modifies fields in the volume catalog record.

AZ: IGGPEXT locates fields in the volume catalog record.

BI: IGGPGET retrieves the volume catalog record.

19 IGGPSALL assigns a specified volume as a candidate volume to a VSAM object.

21.1 IGGPRFDC deletes the RACF profiles for the data sets being deleted, if the request is to force-delete a RACF-indicated catalog.

22 When the caller's request is DELETE CATALOG, IGGPDELC processes it.

AG: IGGPPDE issues PUT-Update to rewrite the record as a free catalog catalog record.

AG: IGGPCCCR checkpoints the catalog record before it is deleted.

A3: IGGPRPLF dequeues the catalog (releases exclusive control over the catalog so that other requests can process its records).

AZ: IGGPEXT locates catalog record fields.

BI: IGGPGET issues GET to retrieve the catalog records.

23 IGGPDBVC verifies that the caller's work area (to return the name of the catalog in) is in the caller's address space.

24 IGGPSCAT searches the catalog being deleted to verify that the catalog is empty (contains no cataloged objects).

Note: If the catalog is identified as a "target" catalog, it can be deleted even though it is not empty. See the discussion on copying a catalog in the "Introduction."

BI: IGGPGET issues GET to retrieve the catalog records that describe the catalog.

25 IGG0CLBU processes the format-4 DSCB:

IGGPF4RD reads the format-4 DSCB.

IGGPF4WR writes the updated format-4 DSCB.

25.1 IGGPRDEL deletes catalog RACF profile.

26 When the caller's request is DELETE SPACE, IGGPDELS processes it.

AG: IGGPPDEC issues PUT-Update to rewrite an extension record as a free catalog record.

AV: IGGPMOD modifies the contents of the volume catalog record's fields.

AZ: IGGPEXT locates a catalog record field.

BI: IGGPGET issues GET to retrieve the volume catalog record.

27 If the volume is completely empty, IGG0CLBU processes the format-4 DSCB:

IGGPF4RD reads the format-4 DSCB.

IGGPF4WR writes the updated format-4 DSCB.

28 When the volume catalog record is not deleted, IGGPLSP determines the amount of available space there is on the volume.

AZ: IGGPEXT locates each shared (nonunique) data space group set of fields (group occurrence) in the volume catalog record.

BI: IGGPGET issues GET to retrieve the volume catalog record.

BV: IGGPSMFL writes SMF record type 69—VSAM Data Space Defined or Deleted.

29 IGGPFDSP scratches all VSAM data space and makes it available to the OS/VS system.

30 IGGPDELB deletes the GDG base catalog record.

Notes for Figure 18 (Continued)

31 IGGPDELX deletes an alias name catalog record and adjusts the alias name catalog record chain. When the deleted alias name catalog record represents the object's only alias, IGGPDELX removes the pointer to the alias name catalog record in the object's catalog record.

32 When the caller's request is DELETE CLUSTER, ALTERNATE INDEX, PATH, PAGESPACE, or NONVSAM, IGGPDEL processes it.

AZ: IGGPEXT locates a catalog record's field.

BI: IGGPGET issues GET to retrieve a catalog record.

33 IGG0CLB5 contains the following DELETE procedures that are called by IGGPDEL:

IGGPDCLS deletes data, index, cluster, and alternate index catalog records.

IGGPDEAX explicitly deletes an alternate index.

IGGPDIAX implicitly deletes one or more alternate indexes.

IGGPDEPT explicitly deletes a path.

IGGPDIPT implicitly deletes one or more paths.

IGGPDUPG deletes upgrade association set of fields.

Each of IGG0CLB5's procedures call the following procedures as needed:

AG: IGGPPDE issues PUT-delete to delete a catalog record.

AV: IGGPMOD modifies the contents of catalog record fields.

AZ: IGGPEXT locates a catalog record field.

BI: IGGPGET issues GET to read a catalog record.

BV: IGGPSMFS writes SMF record type 67 to the SMF data set.

34 IGG0CLA7 contains the following DELETE procedures that are called by IGGPDEL:

IGGPVMSC deletes all space information for the object in the volume catalog record.

IGGPDEMV locates the object's volume information sets of fields (group occurrences).

IGGPDVMV ensures that all required volumes are mounted.

IGGPDUSC returns the data space (issues SCRATCH to delete the format-1 identifier DSCB) from the volume's VTOC.

IGGPMCRA verifies that the primary CRA volume is either mounted or mountable.

IGGPDF4T updates the format-4 DSCB and the volume catalog record timestamp.

Each of IGG0CLA7's procedures call the following procedures as needed:

AG: IGGPPUPC issues PUT-Update to rewrite an updated catalog record.

AV: IGGPMOD modifies the data space group set of fields (group occurrences).

AZ: IGGPEXT locates a catalog record field.

BI: IGGPGET issues GET to read a catalog record.

35 IGGPSSCR returns the space allocated to a catalog or cluster that is contained in a shared (nonunique) data space.

AZ: IGGPEXT locates the data space group set of fields (group occurrence) that describes the data space.

AV: IGGPMOD modifies the data space group set of fields (group occurrence).

36 IGGPGMR adjusts the space map set of fields to show the newly allocated tracks.

37 IGG0CLBU contains the following external procedures used by the DELETE procedures:

IGGPF4RD reads a format-4 DSCB from the VTOC.

IGGPF4DQ issues the VTOC DEQ macro.

IGGPF4WR writes the format-4 DSCB in the VTOC.

38 IGGPGUND cleans up extension records.

AG: IGGPPDEC issues PUT-Update to rewrite an extension record as a free catalog record.

39 IGG0CLB1 contains the following external procedure used by DELETE procedures:

IGGPUDEL deletes upgrade association sets of fields from the Y catalog record.

IGGPUADD adds a non-upgrade alternate index to the upgrade set so that it can be dynamically allocated for CRA processing.

40 IGG0CLA3 contains the following external procedures used by DELETE procedures:

IGGPRPLF releases the serialability of the catalog resource after performing the erase process.

IGGPRPLM acquires the serialability of the catalog resource after performing the erase process.

41 IGG0CLA5 contains the following external procedure used by DELETE procedures:

IGGPDALL deletes the ALIAS chain for nonVSAM catalog records.

42 IGG0CLA4 contains the following external procedure used by DELETE procedures:

IGGPERUG deletes the GDG association sets of fields for a nonVSAM catalog record.

42.1 IGGPRDEL deletes the RACF profile of a primary item.

43 When the caller's request is LISTCAT, IGGPLSTC processes it. IGGPCKAU is called each time a record is retrieved, and verifies the user's authorization to retrieve the record.

AZ: IGGPEXT locates a catalog record field.

BI: IGGPGET issues GET to retrieve the catalog record.

44 When the caller's request is CONVERTV, IGGPCONV processes it.

AG: IGGPPUPC writes the catalog record.

AG: IGGPRCCR updates the catalog control record to indicate the next free control interval.

AH: IGGPSCAT searches the master catalog for the user catalog entry.

AV: IGGPMOD updates device type fields in data and index records in the catalog.

A3: IGGPRPLF releases the master catalog from exclusive control.

BU: IGGPF4RD reads the format-4 DSCB.

BU: IGGPF4WR writes the format-4 DSCB.

BZ: IGGPGALO gets the catalog record whose volume information is to be updated.

BZ: IGGPVALI checks the validity of the CTGPL and CTGFLs.

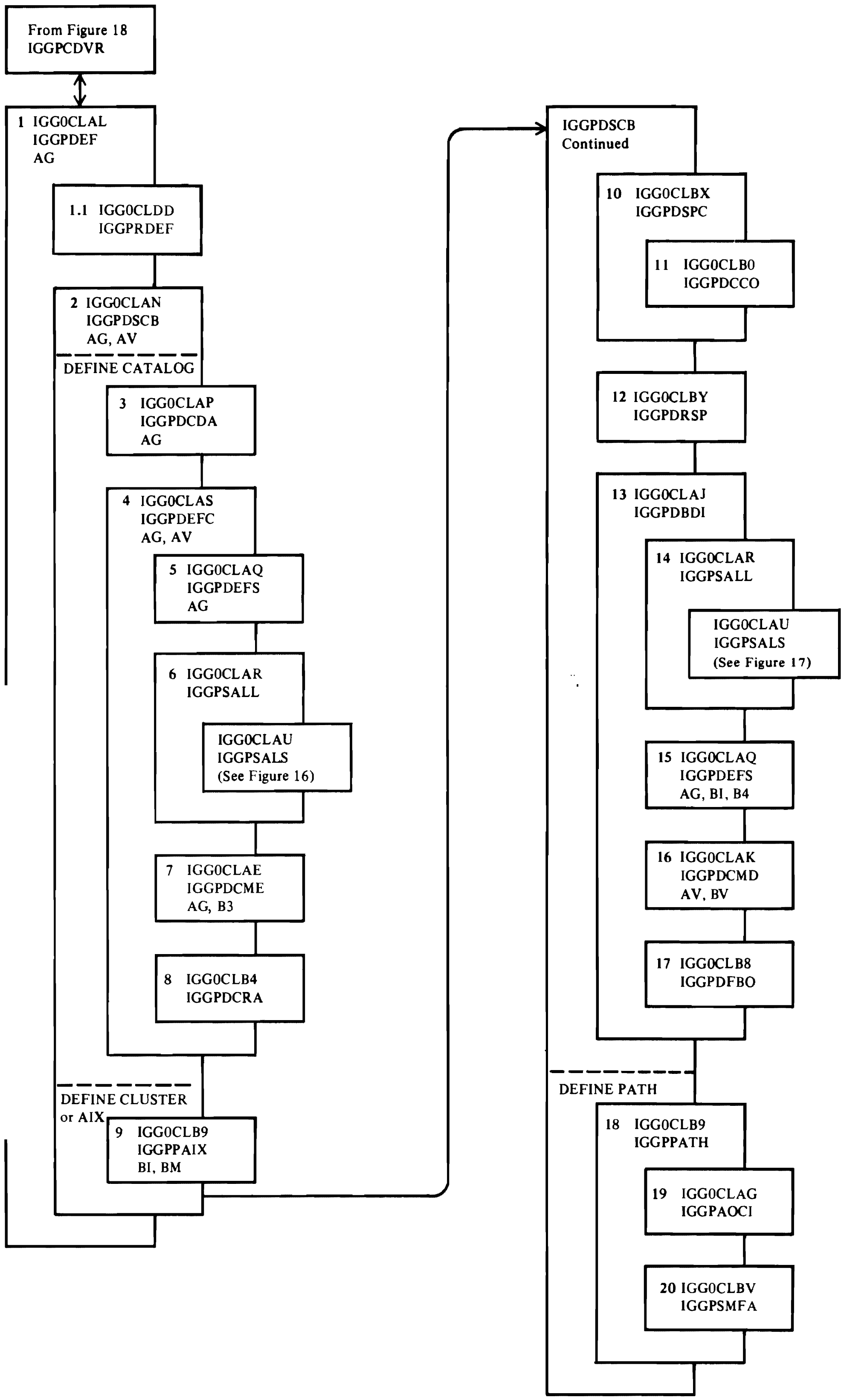


Figure 19. DEFINE Processing

Notes for Figure 19

This figure describes the processing done when a user's program issues DEFINE CATALOG to create a master or users catalog, DEFINE CLUSTER to create a VSAM data set, DEFINE AIX to create an alternate index, or DEFINE PATH to establish an association between a base cluster and one or more alternate indexes. These commands are Access Method Services commands, and they are described in *OS/VS2 Access Method Services*. DEFINE CATALOG is also described in Method of Operation Diagram EE1, DEFINE CLUSTER in Diagram ED1, DEFINE AIX in Diagram ED3, and DEFINE PATH in Diagram ED5.

1 IGGPDEF verifies user-provided values, propagates certain parameters, and generates a name for the data set and index of a cluster, as necessary.

1.1 IGGPRDEF issues the RACDEF SVC (133), if the generated data or index name is to be RACF-protected.

2 IGGPDSCB is a continuation of IGGPDEF.

3 IGGPDCDA checks the validity of user-supplied parameters and determines the default value of other required parameters.

4 IGGPDEFC creates an OS/VS2 user catalog.

5 IGGPDEFS builds a data space. The data space is nonunique (contains parts of more than one VSAM object).

6 IGGPSALL and IGGPSALS assigns part of the data space's tracks to the catalog—a key-sequenced, key-range data set.

7 IGGPDCOC builds a temporary ACB and temporarily opens the catalog so that the catalog's self-describing records (control interval numbers 0 through 13) can be written.

8 IGGPDCRA builds a catalog recovery area if the catalog is recoverable.

9 IGGPPAIX checks the validity of the user-supplied input parameters and the password for the alternate index.

10 IGGPDSPC determines the physical requirements of the object—its control interval size, block size, and number of blocks per track—based on the device that will contain the object.

11 IGGPDCCO checks data and index device characteristics.

12 IGGPDRSP determines the cluster's space allocation quantities.

13 IGGPDBDI partially builds the records required for the data set and index and obtains the primary space allocation for the data set and index.

14 IGGPSALL assigns candidate volumes to the data set and index and, if they reside in a nonunique data space (contains more than one VSAM object), IGGPSALS allocates space from the data space.

15 When the data set or index is to reside in a unique data space (contains only one VSAM object), IGGPDEFS obtains a unique data space for the data set or index.

AG: IGGPISCI ensures that there is a sufficient number of control intervals.

BI: IGGPGET retrieves the volume record.

B4: IGGPDCRA defines a catalog recovery area on the volume if ownership is taken by a recoverable catalog.

16 IGGPDCMD completes the catalog record construction.

AV: IGGPMOD modifies the contents of catalog record fields.

17 If an error occurred during the object's creation, IGGPDFBO resets any allocated tracks to an unallocated status and rewrites any partially built catalog records as free catalog records.

18 IGGPPATH verifies user-supplied input parameters and the password for the path.

19 IGGPAOCI obtains one catalog control interval to contain the path record.

20 IGGPSMFA writes SMF record type 63 after a VSAM path is successfully defined.

Catalog Management I/O Functions

When a catalog record is retrieved, updated (rewritten), written (added to the catalog), or deleted, one of the following catalog management procedures initiates the I/O operation:

- IGGPGET—Retrieves the catalog record
- IGGPPUPC—Updates (rewrites) a catalog record
- IGGPPAD—Adds a record to the catalog
- IGGPPDE—Deletes a catalog record
- IGGPRAG—Retrieves a CRA record
- IGGPRAPU—Updates a CRA record
- IGGPRAPA—Adds a CRA record
- IGGPRAPD—Deletes a CRA record

Before a new catalog record can be written into the catalog, a catalog's control interval is assigned to contain the new record's information. This assignment is made by:

- IGGPAOCI—assigns more than one contiguous (if possible) control interval to the caller. This function is usually called during a DEFINE procedure.
- IGGPAXCI—assigns one control interval to the caller. This function is usually called when an extension record is being built.

The above-mentioned procedures call other catalog management I/O procedures to perform special functions. The other procedures include IGGPCCCR, IGGPISCI, IGGPRAEA, IGGPRAOP, IGGPRAOR, IGGPRARA, IGGPRASC, IGGPRAX, IGGPRCCR, IGGPXIO, and IGGPXRIO.

This section contains a detailed explanation of the catalog management I/O procedures.

all catalog and catalog recovery area I/O functions are described in Figures 20 to 38.

Catalog Management RACF Functions

The procedures within module IGG0CLDC are used to check a user's authorization to gain access to data sets or catalog entries, if these resources are protected by RACF. Authorization to use a RACF-protected entity is accomplished by issuing the RACHECK macro instruction. The procedures within module IGG0CLDD provide routines to alter the RACF indicator for a VSAM object and to build, alter, or delete the RACF profiles by means of the RACDEF macro instruction. The RACDEF and RACHECK macros are described in *OS/VS2 MVS Resource Access Control Facility (RACF) General Information Manual.*

Figures 38.1 through 38.9 describe the catalog management RACF functions.

IGGPGET—catalog record retrieval

- Retrieves the requested control interval and places it into the caller-specified record area.
- Initializes the RAB (one of six record area blocks in the CCA) and its record segment pointers.

Normal keyed GET

1. IGGPXIO retrieves the true name record, using keyed direct GET.
2. IGGPGET converts the catalog record's control interval number (in the true name catalog record) to an RBA.
3. IGGPXIO retrieves the catalog record's control interval, using addressed direct GET.

Normal keyed GET-next

– Same as "normal keyed GET", except that "key greater than or equal" is specified when IGGPXIO retrieves the true name catalog record.

Normal GET by control interval number

1. IGGPGET converts the catalog record's control interval number (specified by the caller) to an RBA.
2. IGGPXIO retrieves the catalog record's control interval, using addressed direct GET.

GET requests issued while a catalog is being opened or created

– When the catalog is being created, IGGPGET "retrieves" the record by locating the DEFINE workarea that contains the requested record.

– When the catalog is being opened, IGGPGET issues EXCP and WAIT to retrieve the requested record.

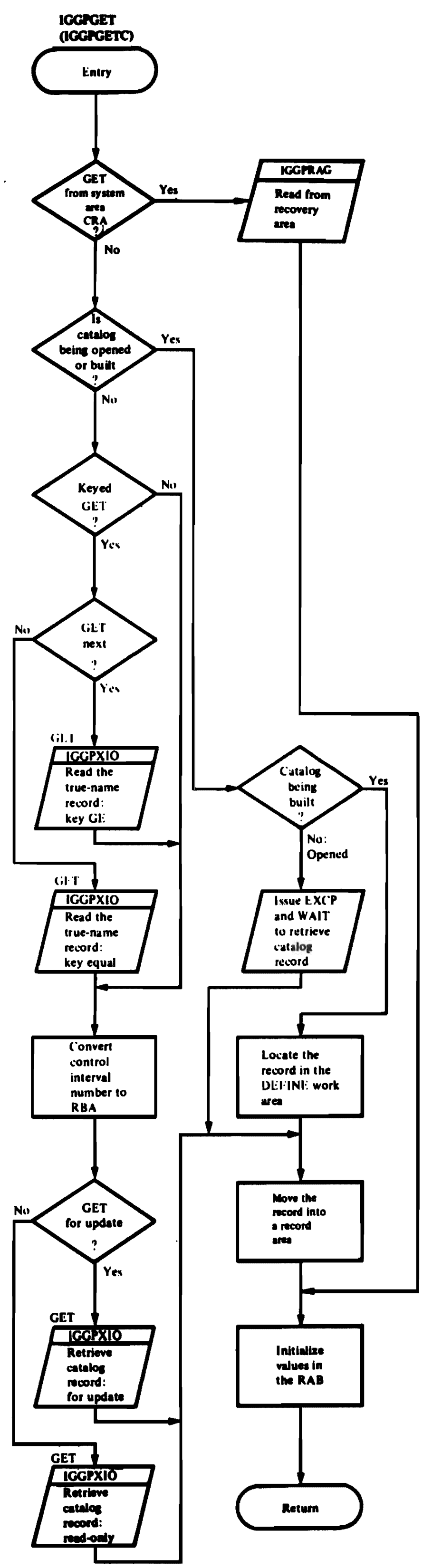


Figure 20. Retrieve a Catalog Record (IGGPGET)

IGGPPUPC—catalog record write (update)

1. If the RPL used to retrieve the updated record has been reused, IGGPTRPL calls IGGPXIO to retrieve the catalog record again. This verifies the catalog record's position in the catalog and re-establishes the RPL's GET-for-update status.
2. IGGPXIO writes the catalog record into the catalog, using addressed direct PUT-for-update.

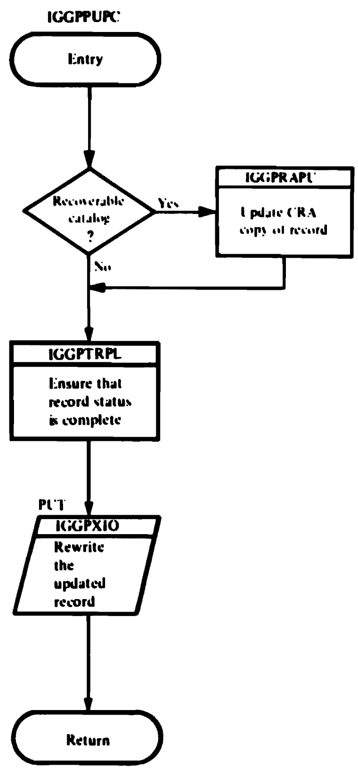


Figure 21. Write (Update) a Catalog Record (IGGPPUPC)

IGGPPAD—catalog record write (add)

1. IGGPXIO retrieves the free control interval that will be replaced by the new record, to verify the control interval's position in the catalog and to re-establish the RPL's GET-for-update status.
2. IGGPXIO writes the new catalog record into the catalog, using addressed direct PUT-for-update.
3. If a true name catalog record is required, IGGPPAD builds it, then calls IGGPXIO to write it into the catalog using keyed direct PUT.

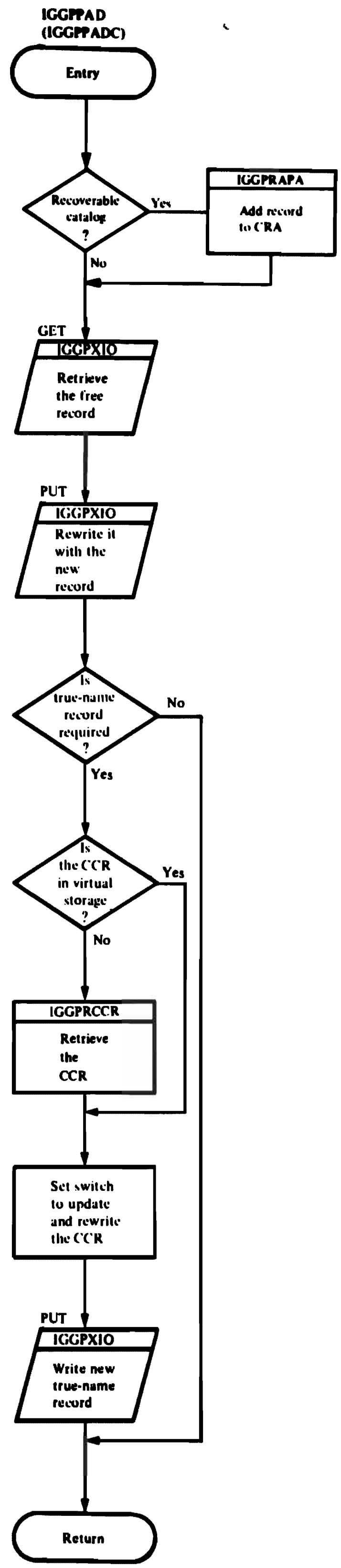


Figure 22. Write (Add) a Catalog Record (IGGPPAD)

IGGPPDE—catalog record deletion

1. If a true name catalog record exists for the record being deleted:
   - IGGPXIO retrieves the true name catalog record, using keyed direct GET, to establish the record's GET-for-update status.
   - IGGPXIO erases the true name record, using keyed ERASE.
2. If the catalog control record (CCR) is not already in a catalog management record area, IGGPRCCR retrieves it.
3. If the RPL used to retrieve the record (to be deleted) has been reused, IGGPTRPL calls IGGPXIO to retrieve the catalog record again to verify the record's position in the catalog and to re-establish the RPL's GET-for-update status.
4. IGGPPDE builds a free catalog record. Its control interval number is the same as the record to be deleted. The control interval number of the free catalog record is put into the catalog's free-control-interval chain.
5. IGGPXIO deletes the record by replacing it with the free catalog record, using addressed direct PUT-for-update.
6. IGGPCCCR updates and rewrites the catalog control record (CCR).

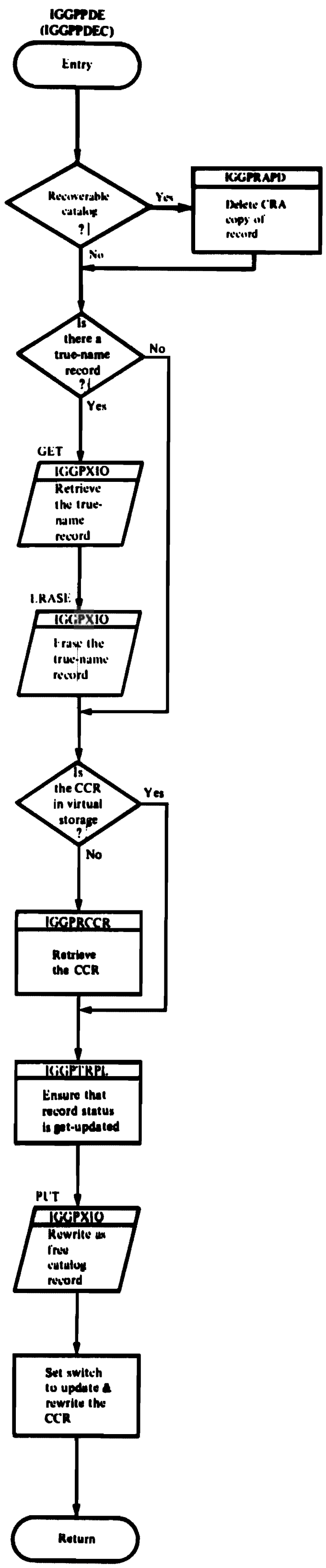


Figure 23. Delete a Catalog Record (IGGPPDE)

IGGPAOCI—assigns contiguous (if possible) catalog control intervals to the caller

1. If the catalog control record (CCR) is not already in a catalog management record area, IGGPRCCR retrieves it.
2. IGGPAOCI assigns the requested number of catalog control intervals to the caller. The control intervals will be used to contain new catalog records (built during a DEFINE process). One of the following methods is used (in the order listed) to obtain the control intervals:

   A. If enough previously unassigned control intervals are available in the catalog's extent, IGGPANCI preformats and assigns the requested number of control intervals to the caller.

   B. If enough free control intervals (not necessarily contiguous) are available, IGGPAOCI removes the requested number of control intervals from the free control interval chain and assigns them to the caller.

   C. IGGPANCI obtains more space via catalog extend for the catalog, then preformats and assigns the requested number of control intervals to the caller.
3. IGGPCCCR updates and rewrites the catalog control record (CCR).

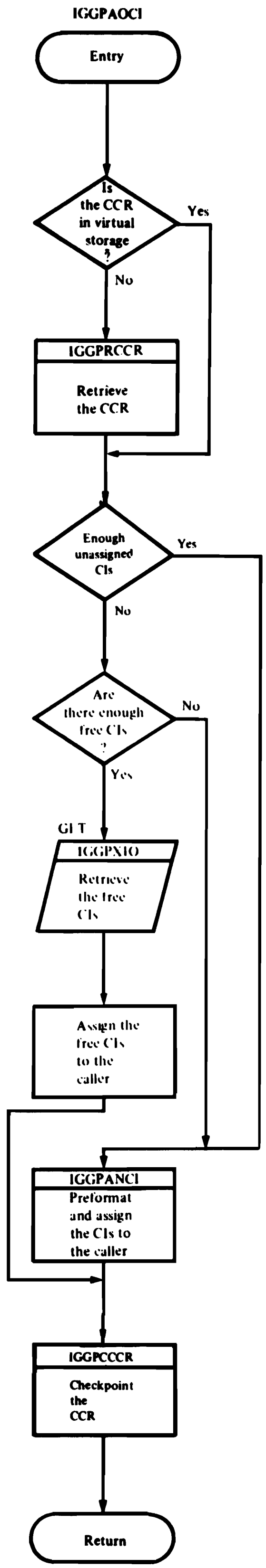


Figure 24. Assign Catalog Control Intervals to the Caller (IGGPAOCI)

IGGPAXCI—assigns one catalog control interval to be used as an extension record

1. If the catalog control record (CCR) is not already in a catalog management record area, IGGPRCCR retrieves it.
2. IGGPAXCI assigns one control interval to the caller. One of the following methods is used (in the order listed) to obtain the control interval:
   A. If a free control interval is available, IGGPAXCI removes it from the free control interval chain and assigns it to the caller.
   B. IGGPANCI preformats and assigns a previously unassigned control interval to the caller. This might mean that IGGPANCI first obtains more space for the catalog, via catalog extend.
3. IGGPCCCR updates and rewrites the catalog control record (CCR).

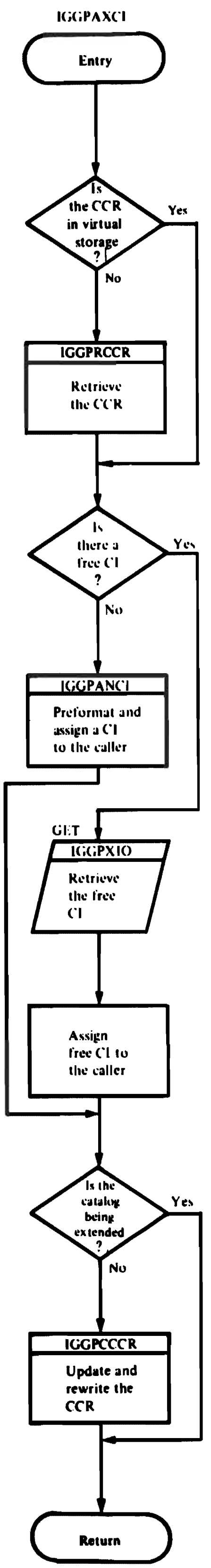


Figure 25. Assign a Catalog Control Interval for an Extension Record (IGGPAXCI)

IGGPCCCR—updates and rewrites the catalog control record (CCR)

1. The CCR is read to establish its set-for-update status.
2. IGGPCCCR updates the CCR control fields to reflect the current catalog record usage.
3. IGGPCCCR calls IGGPXIO to rewrite the CCR, using addressed PUT-for-update.

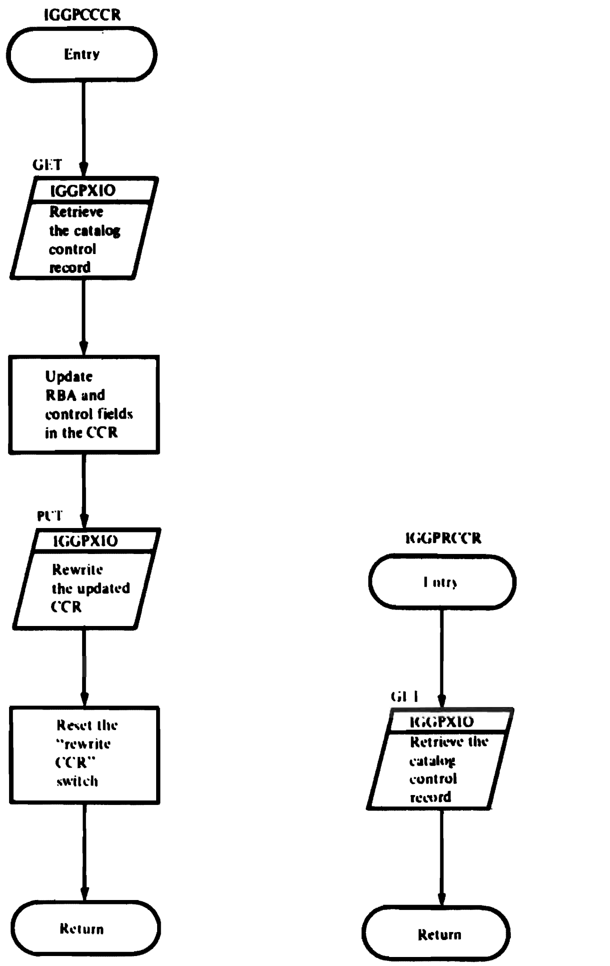


Figure 26. Update and Rewrite the CCR (IGGPCCCR)

IGGPRCCR—reads the Catalog Control Record (CCR) and updates control fields and RBAs from information contained within the CCR

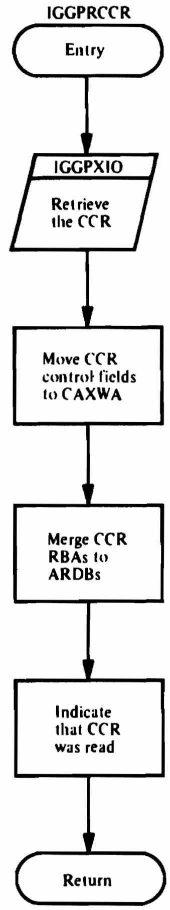


Figure 27. Read the CCR and Update Control Fields and RBAs (IGGPRCCR)

IGGPXIO—calls VSAM record management

1. IGGPXIO initializes an RPL.
2. IGGPXIO issues GET, PUT, or ERASE—a VSAM record management request macro instruction.
3. If an error occurs, IGGPXIO exits on error to IGGPIORA to convert the RPL error code to an appropriate catalog management error code. IGGPIORA then returns to the routine that called the catalog I/O function that was processing when the error occurred.

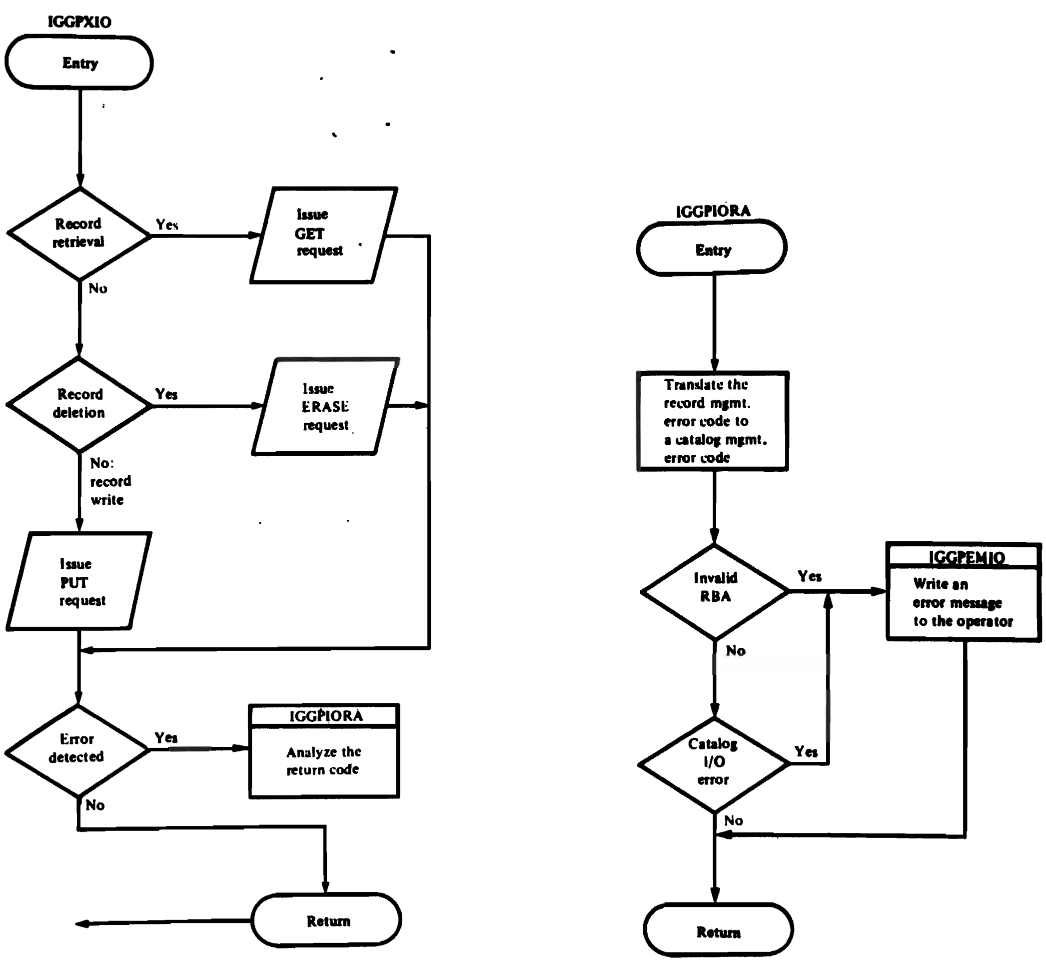


Figure 28. Call VSAM Record Management for Catalog Request (IGGPXIO)

IGGPISCI—ensures that there are enough free control intervals, so that the catalog won't be extended (obtain more space) at the same time a volume catalog record is being processed

1. If the catalog control record (CCR) is not already in a catalog management record area, IGGPRCCR retrieves it.
2. If there are not sufficient free control intervals available, IGGPISCI calls IGGPANCI to obtain more space (via catalog extend) for the catalog. IGGPANCI then preformats and assigns a previously unassigned control interval to IGGPISCI.
3. IGGPXIO retrieves the control interval to establish its GET-for-update status, using addressed GET-for-update.
4. IGGPXIO rewrites the control interval as a free control interval, using addressed PUT-update.
5. IGGPCCCR updates and rewrites the catalog control record (CCR).

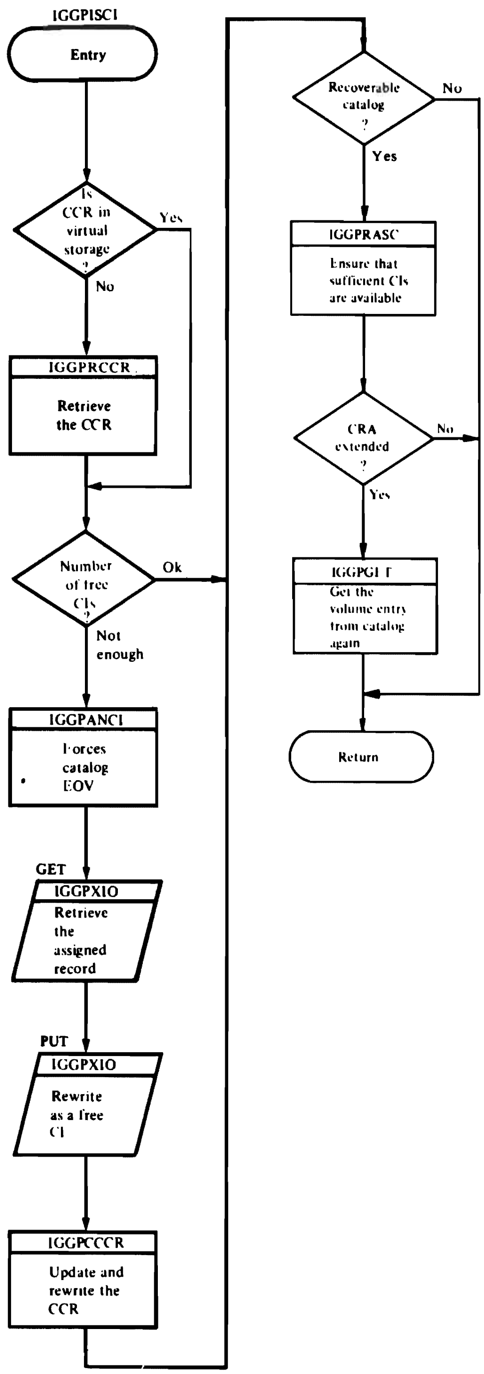


Figure 29. Ensure Availability of Catalog Control Intervals (IGGPISCI)

IGGPRAPU—CRA record write (update)

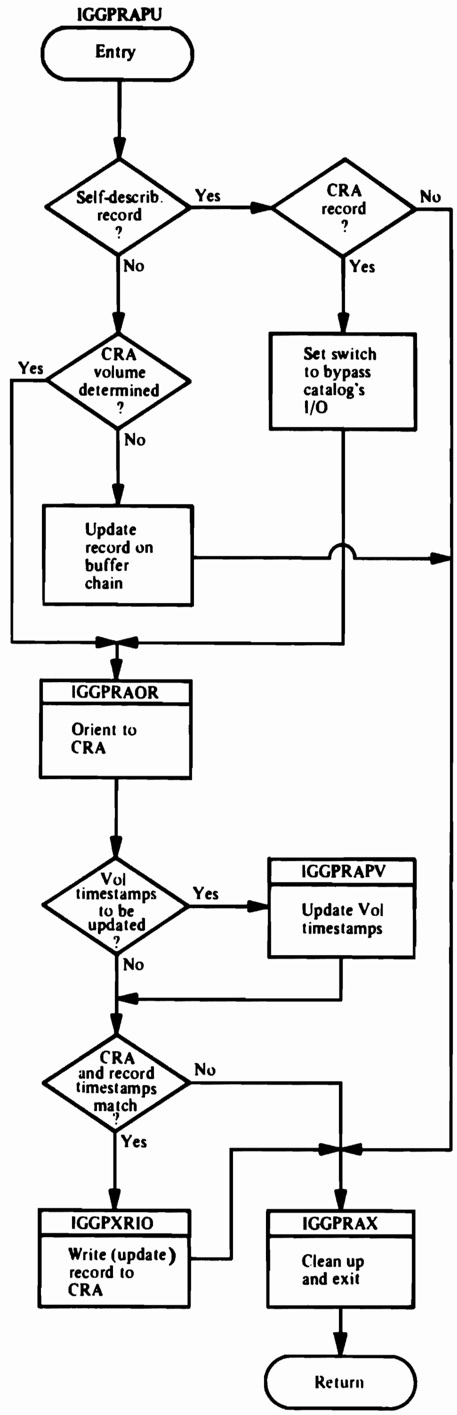


Figure 30. Write (Update) a CRA Record (IGGPRAPU)

IGGPRAPA—CRA record write (add)

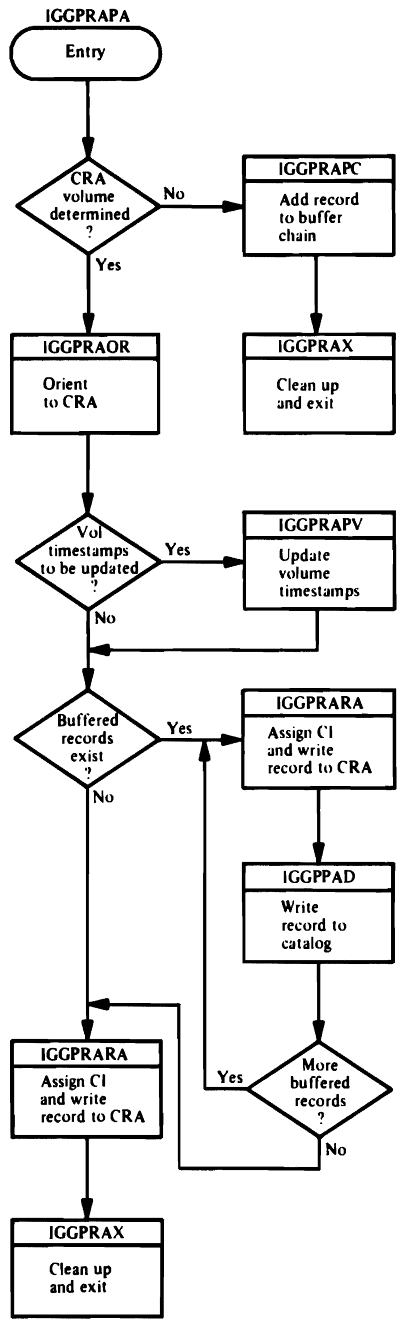


Figure 31. Write (Add) a CRA Record (IGGPRAPA)

IGGPRAPD—CRA record delete

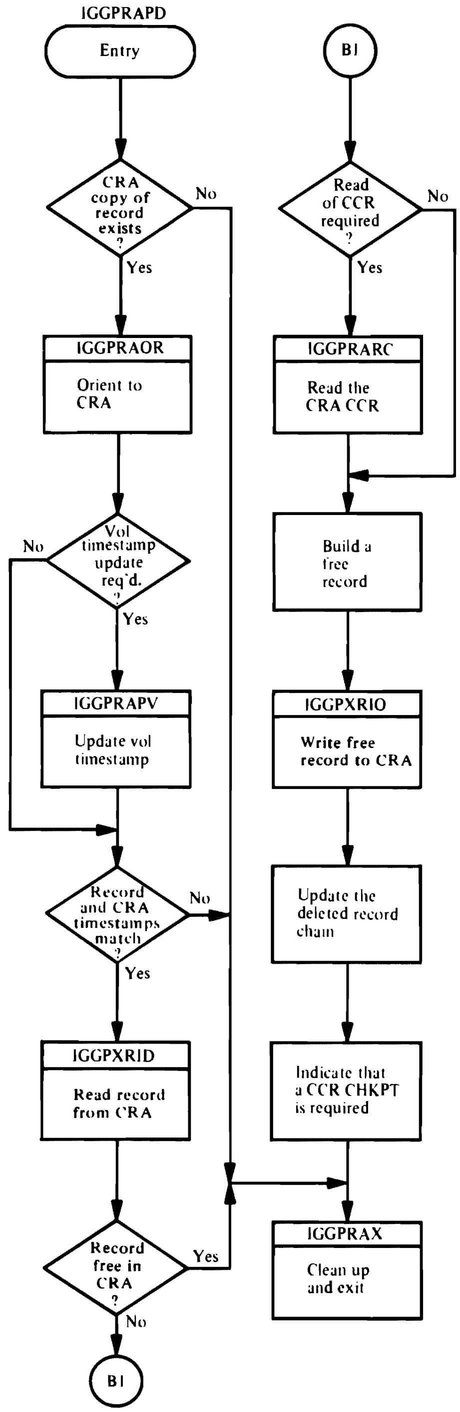


Figure 32. Delete a CRA Record (IGGPRAPD)

IGGPRAOR—orient to CRA

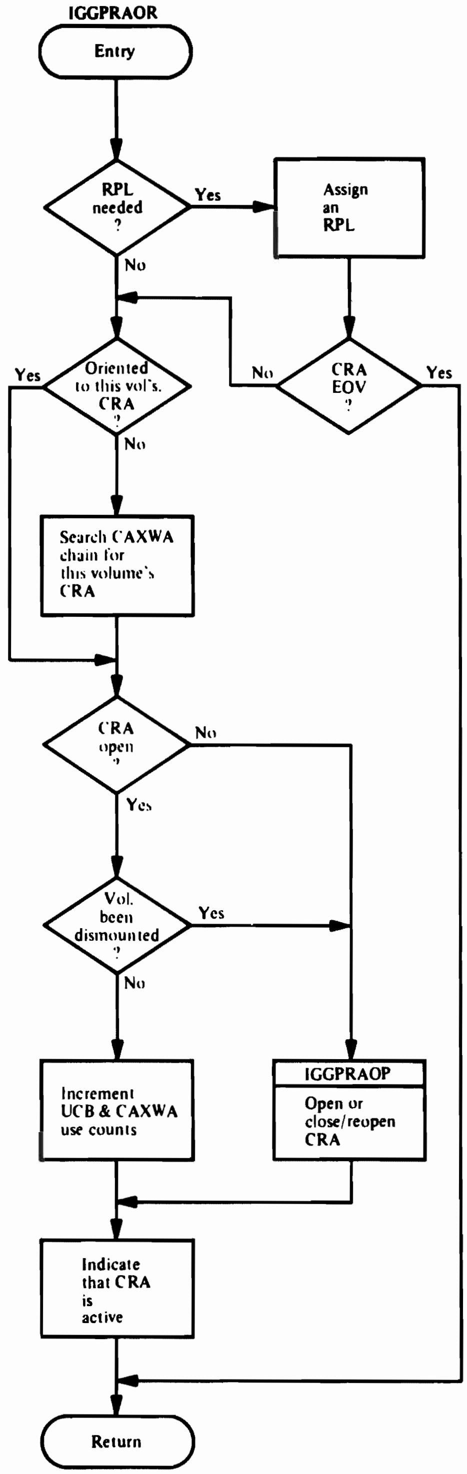


Figure 33. Orient to the CRA (IGGPRAOR)

IGGPRAOP—CRA OPEN

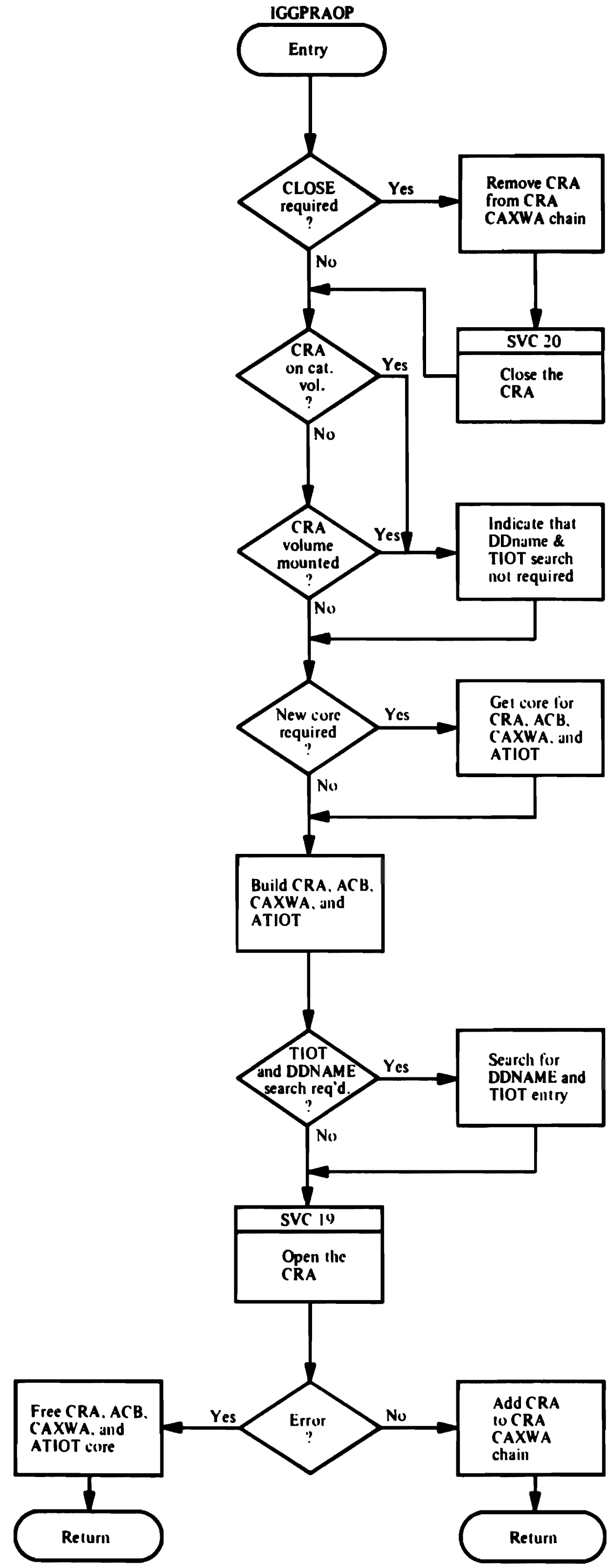


Figure 34. Open a CRA (IGGPRAOP)

IGGPRARA—assign control interval numbers to new CRA records and write them to the CRA

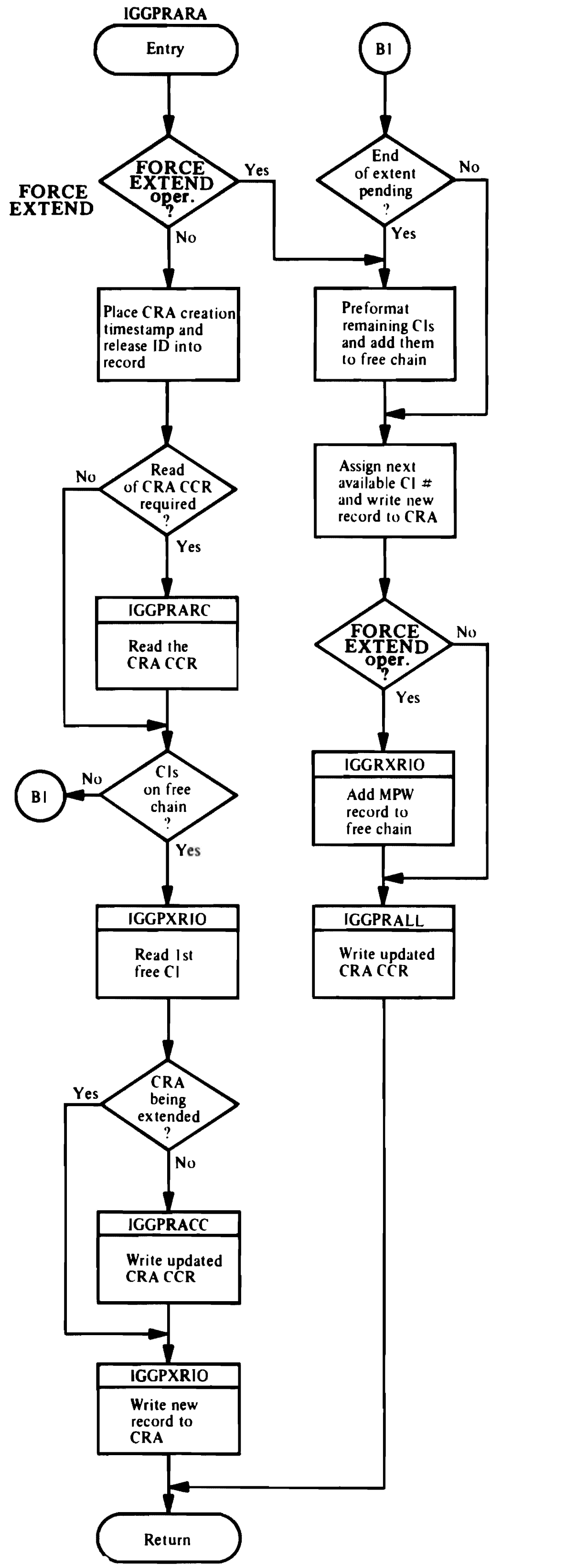


Figure 35. Assign Control Interval Numbers to CRA Records (IGGPRARA)

IGGPRASC—ensures that there are enough free control intervals in a CRA so that the CRA won't be extended at the same time the CRA's volume catalog record is being extended

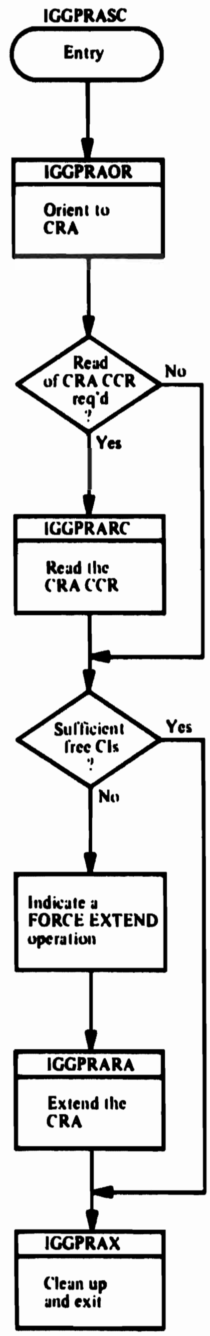


Figure 36. Ensure Availability of CRA Control Intervals (IGGPRASC)

IGGPRAX—CRA I/O function exit routine

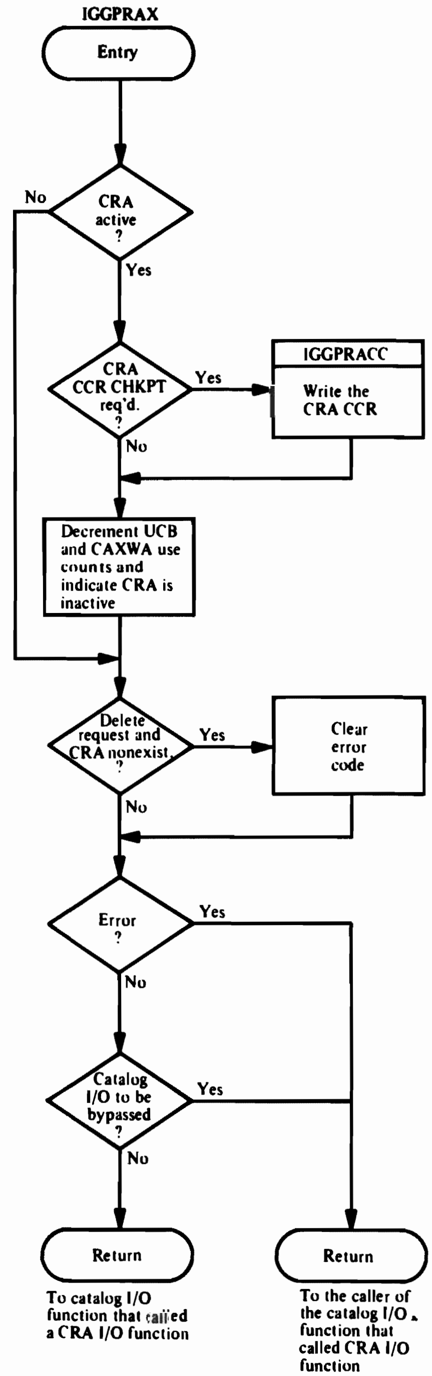


Figure 37. Return from CRA I/O Function (IGGPRAX)

IGGPXRIO—calls VSAM record management

1. IGGPXRIO initializes an RPL.
2. IGGPXRIO issues GET or PUT—a VSAM record management request macro instruction.
3. If an error occurs, IGGPXRIO returns on error to IGGPRAEA to convert the RPL error code to an appropriate catalog management error code.

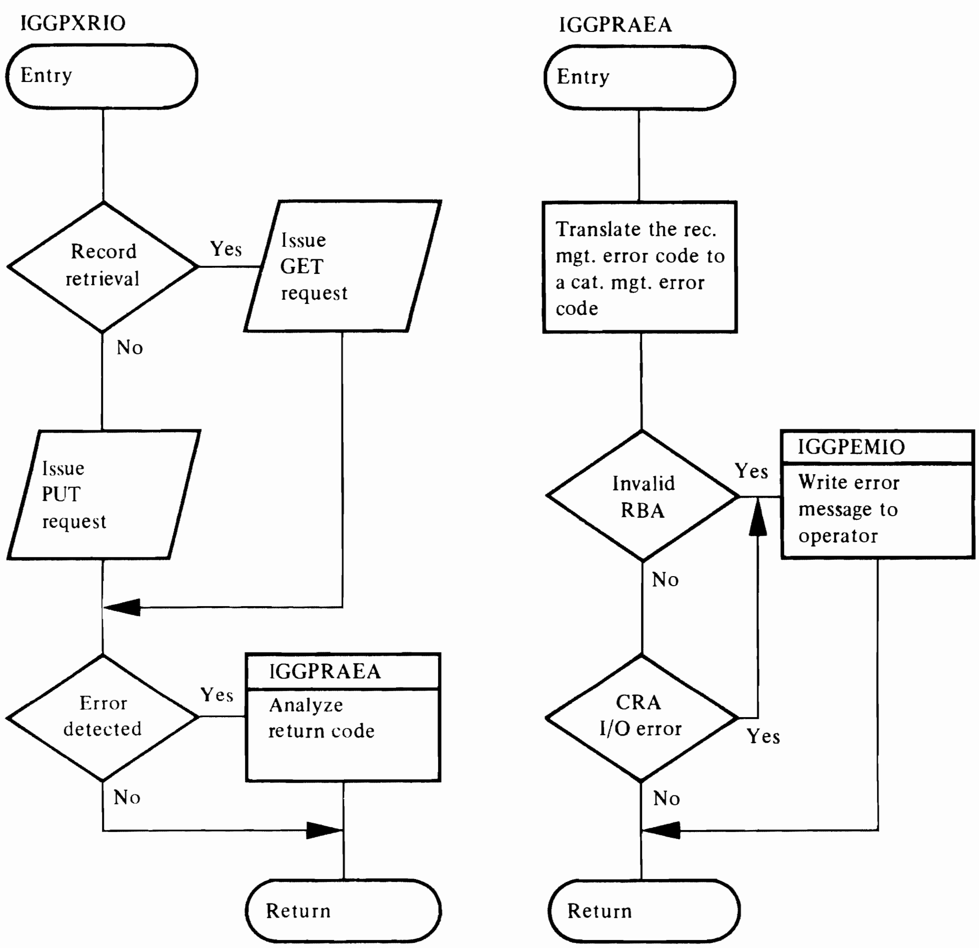


Figure 38. Call VSAM Record Management for CRA Request (IGGPXRIO)

IGGPALRI—Alter RACF Indicator

IGGPALRI is invoked by IGGOCLAT before regular alter processing, if CTGFVRON or CTGFVROF is on.

1. Use TESTAUTH to ensure that the caller is authorized (state, key, or APF).
2. Set or clear the RACF indicator in the catalog entry according to CTGFVRON and CTGFVROF.
3. If a catalog is being altered, set or reset the CAXRAC bit according to CTGFVRON and CTGFVROF.

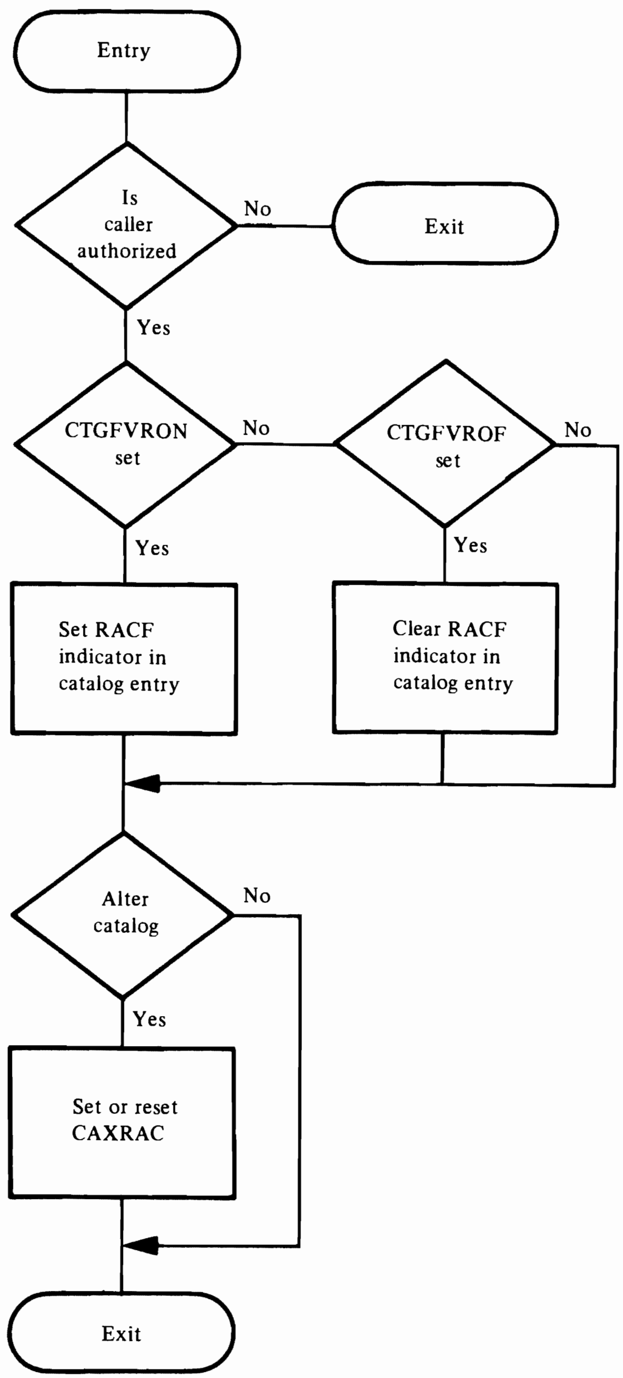


Figure 38.1 Alter RACF Indicator (IGGPALRI)

IGGPALRN—Alter Rename

IGGPALRN is called by IGG0CLBD during alter rename processing.

1. The data set type (RCDID) must be C,D,G,I, or R.
2. If CNAME (the entry name of the catalog record in the current record area) equals the name pointed to by CTGFVENT, the initial rename should be done. CTGFVENT points to the old name and CTGNEWNM points to the new name.
3. If CNAME does not equal the name pointed to by CTGFVENT, then this is a back-out request. CTGNEWNM points to the "old" name and CTGFVENT points to the "new" name.

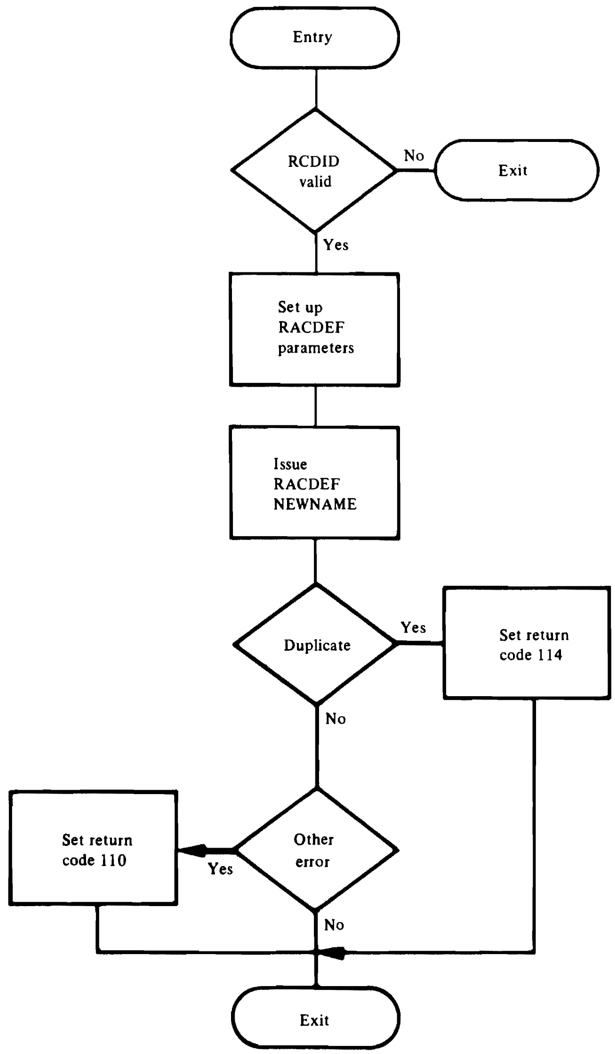


Figure 38.2 RACF Alter Rename (IGGPALRN)

IGGPRDEF—RACF Define

IGGPRDEF is called by IGG0CLAT before defining a VSAM data set or catalog. It is also called by IGG0CLAL after generating a data or index name. In both cases, this routine is not called, if both JSCBADSP and CTGFVRON are off. The address of the FVT is passed in CMSWARFV.

1. If the primary object being defined is a cluster or alternate index, do RACF-define processing for user-supplied data and index names as well as the primary object name, unless the object is a catalog or path (CTGTYPE is M,U, or R).
2. If a catalog is being defined, the volume serial for the RACDEF should come from the FVT's volume list rather than from the CAXWA/UCB.
3. If RACDEF fails because definition of the data set is not allowed by the security subsystem, then exit with return code 118 and set the reason code according to the object being processed at the time.
4. If RACDEF fails for other reasons, exit with return code 114.
5. Save the primary and component names in CMSROBJN (in the CMS work area) as they are RACF-defined.

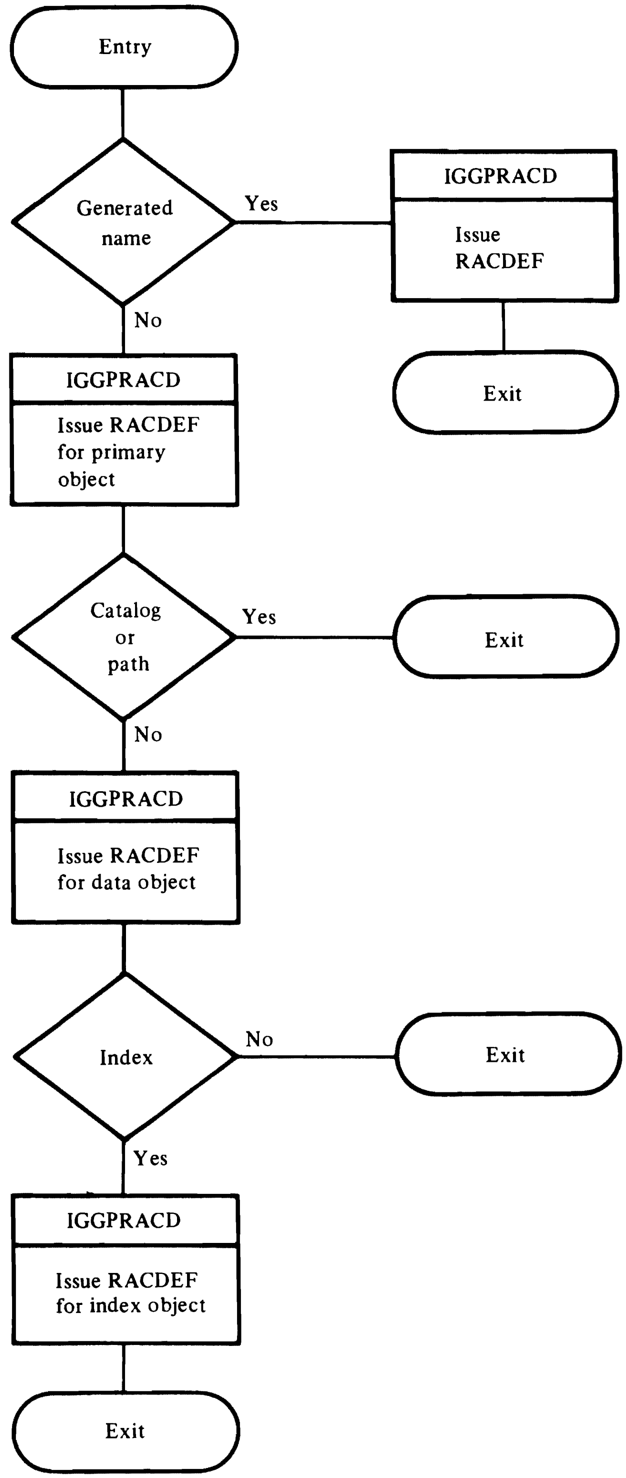


Figure 38.3 RACF Define (IGGPRDEF)

IGGPRDLC—RACF Delete of a Partially-Defined Cluster

IGGPRDLC is called by IGG0CLAT after a define of a VSAM RACF-protected data set has failed. The routine is called for every RACF-indicated object name associated with the failing define.

1. The primary and component names (if any) are found in the CMSROBJN array (in the CMS work area). For a catalog (CTGTYPE is M or U), the data and index names are not defined to RACF. Also for a catalog, the volume serial for the RACDEF is from the FVT's volume list.

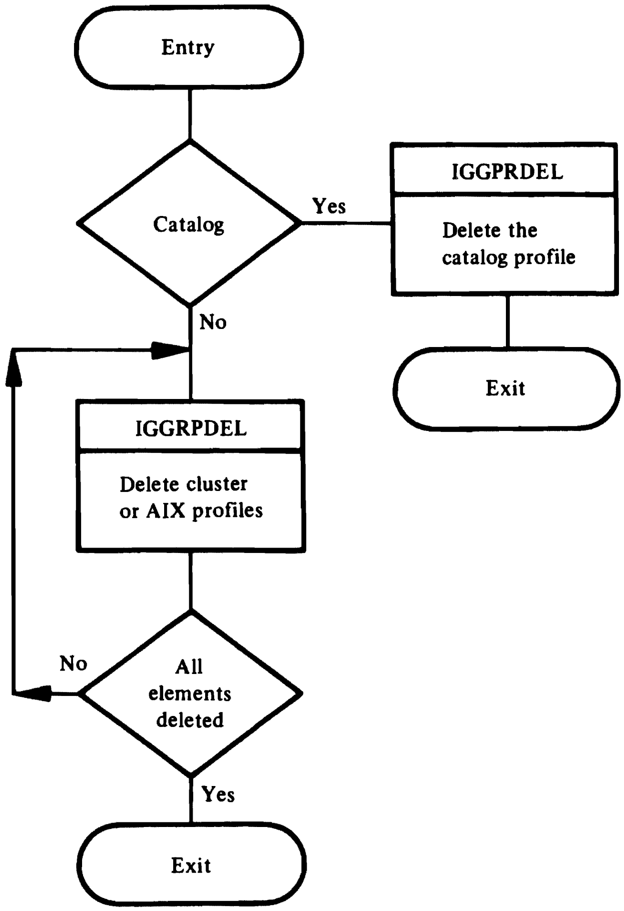


Figure 38.4 RACF Delete of a Partially-Defined Cluster (IGGPRDLC)

IGGPRFDC—RACF Force Delete Catalog

IGGPRFDC is called by IGG0CLAT before a force delete of a catalog.

1. IGGPGALO (in IGG0CLBZ) reads each entry in the catalog except the first. When the low-key range is exhausted, clear the end-of-data condition out of CCAPROB.
2. If the data set described in the entry is RACF-protected, call IGGPRDEL to delete the profile.

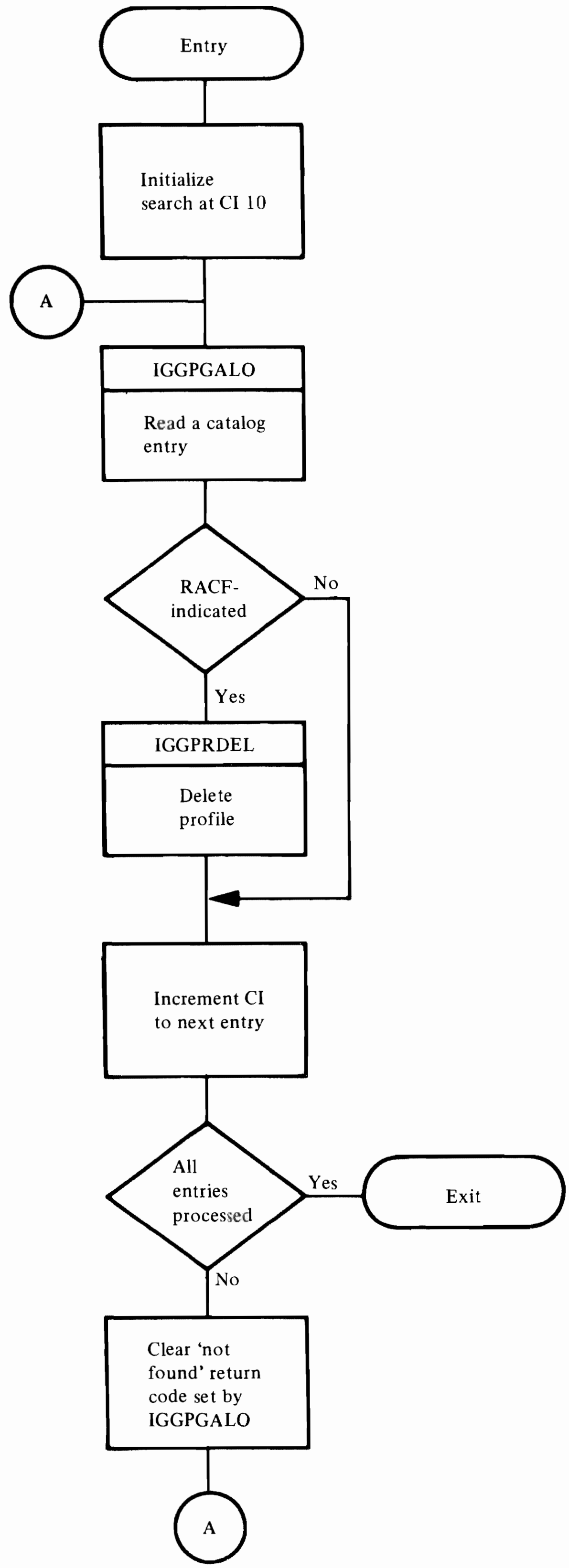


Figure 38.5 RACF Force Delete Catalog (IGGPRFDC)

IGGPRACT—RACF Catalog Verification

IGGPRACT is called by IGGPPWVR (in IGG0CLBM).

1. The catalog entries for items that have implicit protection are GDG base, nonVSAM, user catalog connector, or alias.
2. Global verification indicates that RACF authorization checking is being done through a secondary source because the primary source is not RACF-protected or RACF authorization could not be obtained through the primary source.
3. LISTCAT and locate can be performed with read authorization by suppressing passwords. Two-level checking indicates that, if alter fails, the saved level will then be tried.

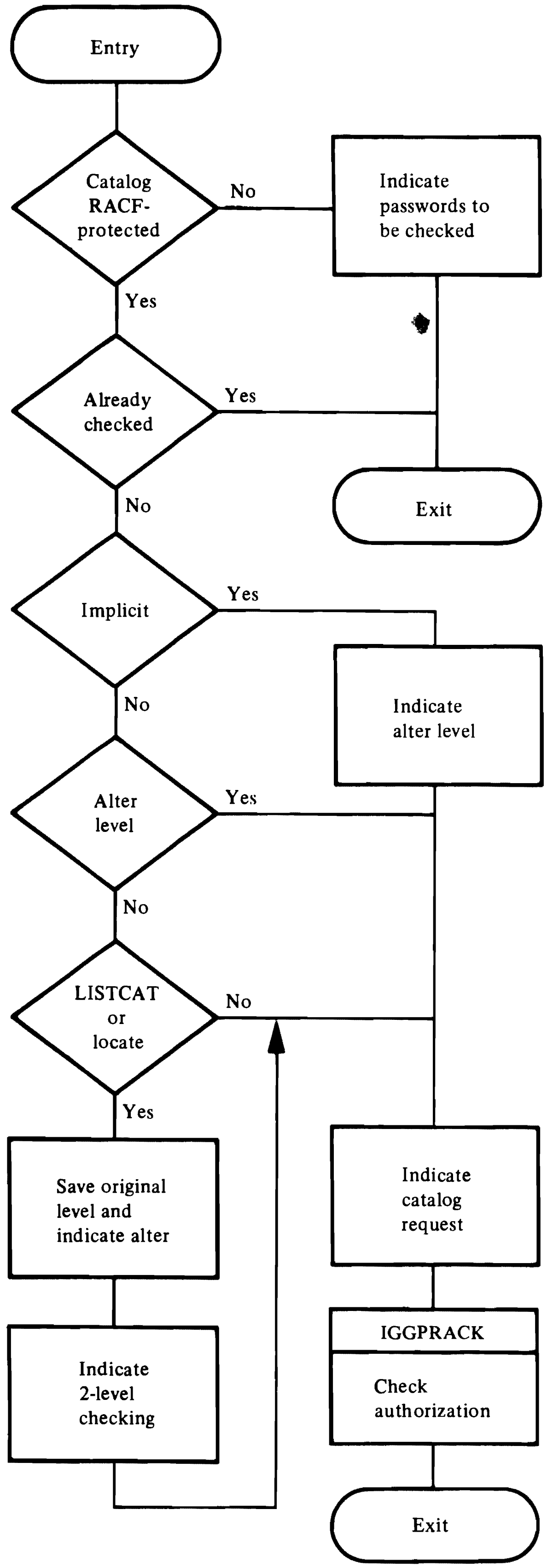


Figure 38.6 RACF Catalog Verification (IGGPRACT)

IGGPRACG—RACHECK for Implicit Alter GDG

IGGPRACG is called by IGG0CLA4 to check authorization to alter the GDG base when defining a GDG generation.

1. Before this routine is called, DEFINE processing will have determined that the user has RACF update authorization to the catalog.

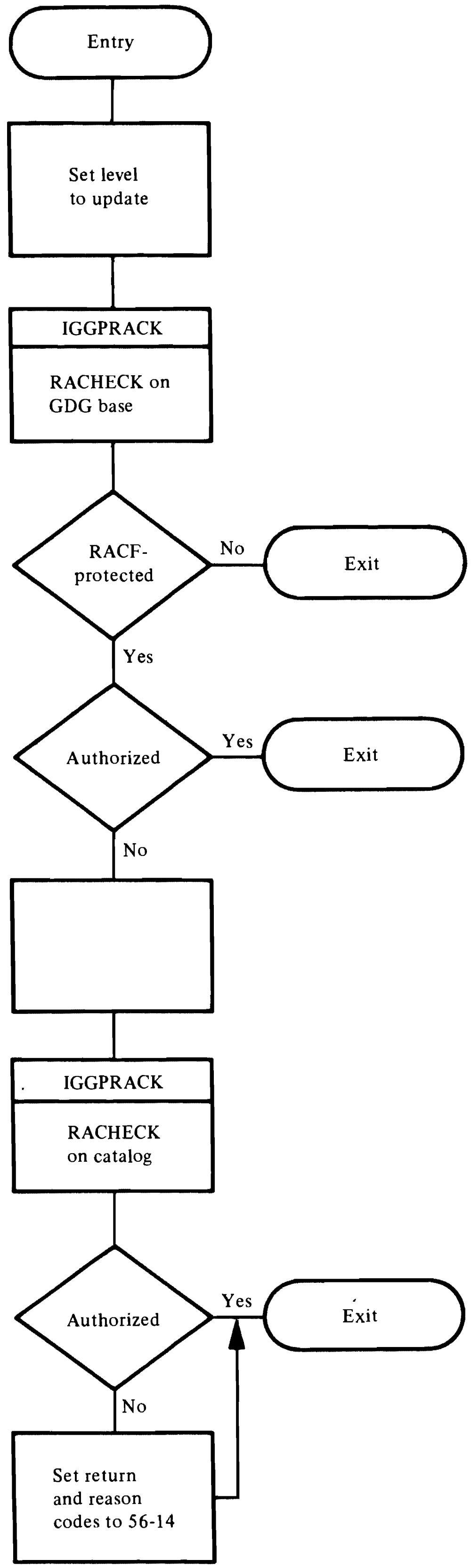


Figure 38.7 RACHECK for Implicit Alter GDG (IGGPRACG)

IGGPRACV—RACF Verification Routine

IGGPRACV is called by IGGPPWVR (in IGG0CLBM) in its initial authorization attempt. If authorization is not obtained by calling this procedure, IGGPPWVR attempts authorization through a global entity.

1. The catalog entries for items that can have implicit protection cannot be RACF-protected or contain passwords. These items are GDG base, nonVSAM, user catalog connector, or alias. This routine is not called for GDG base authorization checking when a GDG generation member is being defined. See IGGPRACG.
2. If 2-level authorization checking is specified, IGGPRACK first tries alter authorization. If that fails, IGGPRACK issues a RACHECK for the saved level.

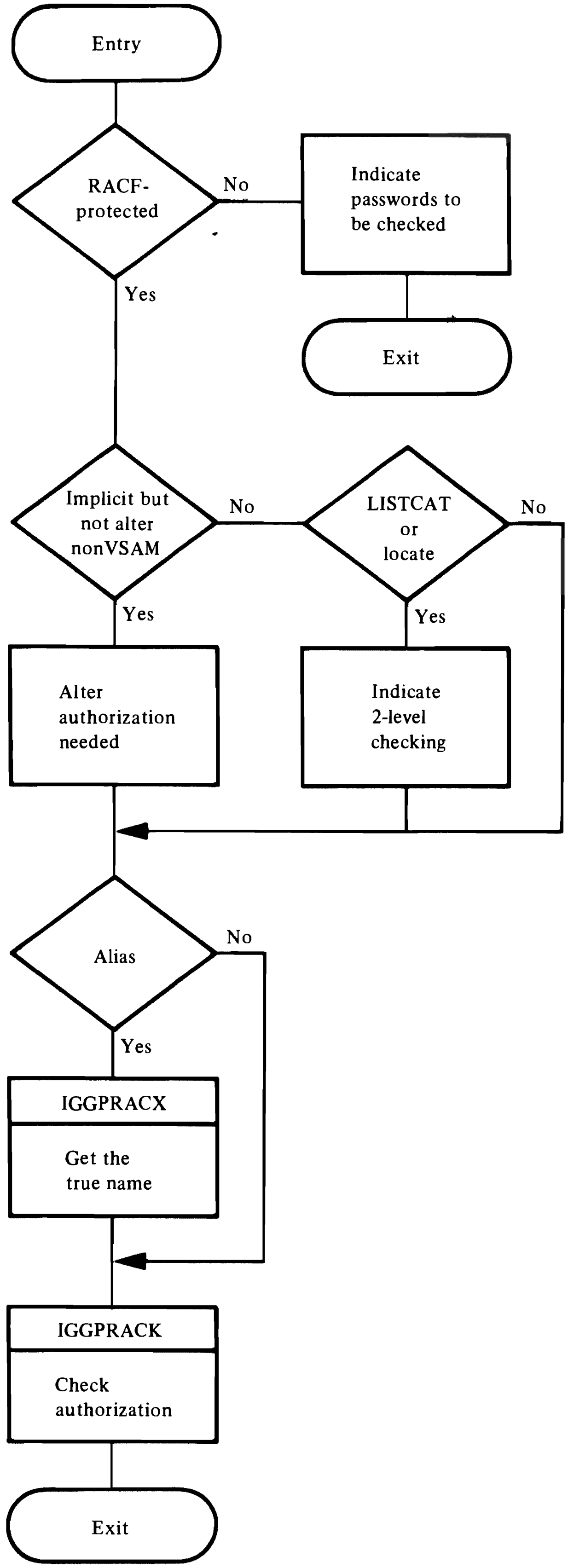


Figure 38.8 RACF Verification Routine (IGGPRACV)

IGGPRACK—RACF Check Routine

IGGPRACK issues RACHECK SVC (130) to determine whether user has RACF authorization.

Parameters passed when IGGPRACK issues RACHECK are the dsname, volser, authorization level, logging option, and the profile option. Except for a user catalog connector and a nonVSAM entry, the volser is that of the volume containing the catalog entry. For a nonVSAM entry, the volser is that of the first volume containing the data set. For a user catalog connector entry, the volser of the volume pointed to by the user catalog connector is used. This is extracted from the volume group occurrence field.

1. If the request is for a global entity (CKAUGLOB is on) then set up to request LOG=NOFAIL, which indicates that authorization failures will not be logged.
2. Issue RACHECK with the authorization level indicated by CKAURACL.
3. If the RACHECK succeeded, set CKAUVFST to indicate verification should stop unless CKAUDOUB is on and CKAURACL is at the non-master level, in which case set CKAUVFRD. If it was the master level that was verified, set CKAUVFMT.
4. If the RACHECK indicates the entity is not defined, indicate by setting CCAPROB non-zero; otherwise, set CCAPROB to zero.
5. If RACHECK failed and CKAUDOUB is on (indicating a double RACHECK is in order), set CKAURACL to the original authorization level and loop back to reissue the RACHECK, unless CKAUVFRD is set, in which case exit. If CKAURACL is at the read level, the two-pass loop will be exited.

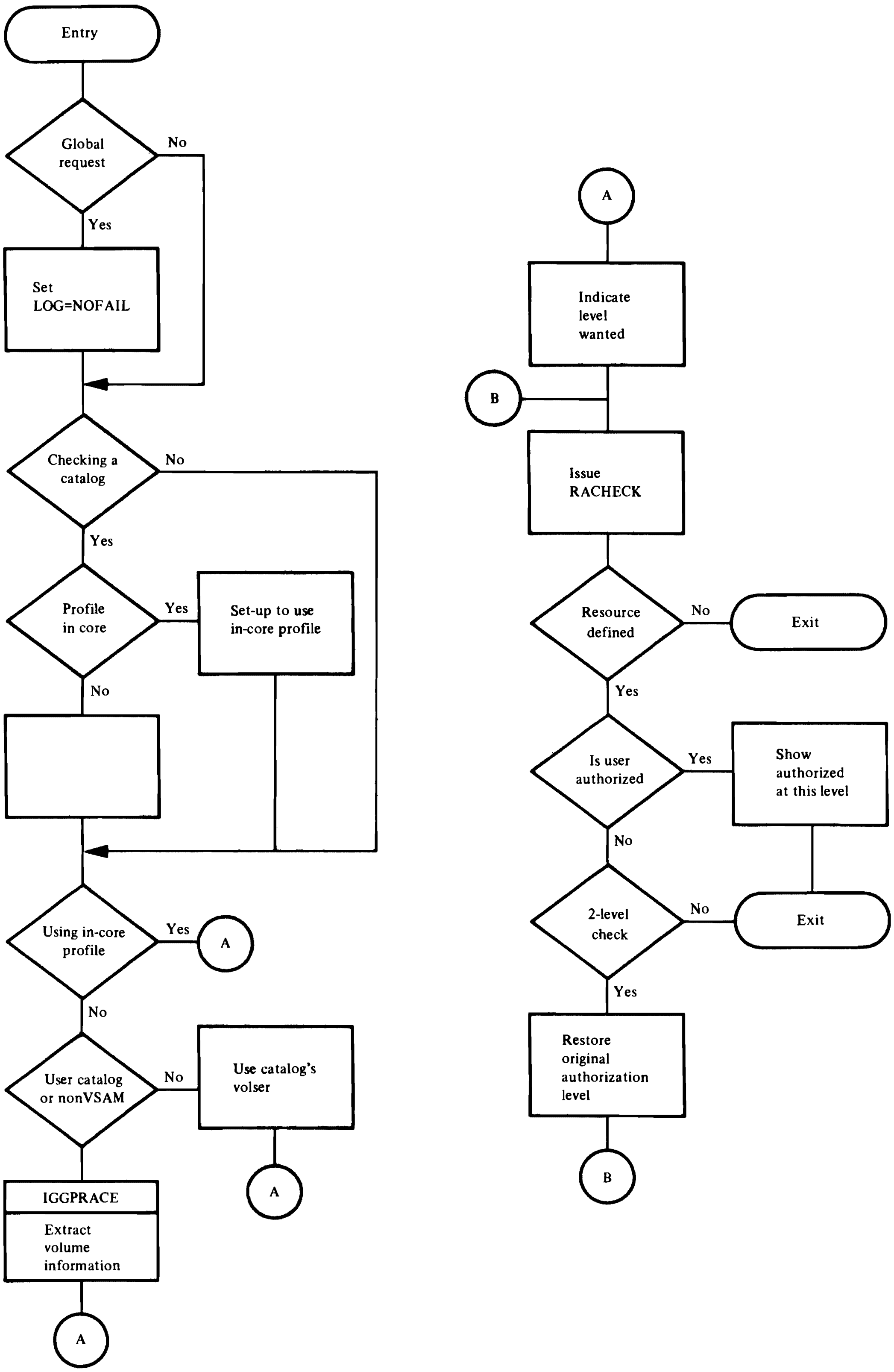


Figure 38.9 RACF Check Routine (IGGPRACK)

DIRECTORY

The directory section contains lists of items and cross-reference information related to each item in the list. The lists include:

- Module Directory
- Module Packaging
- External Procedure Directory
- Procedure Calls Directory
- Procedure Called-By Directory

Module Directory

The module directory is organized alphabetically by symbolic module name. It lists the descriptive name, the module's procedure names (external entry points), and the method of operation diagrams and program organization figures in which that module is referenced.

Module Name	Descriptive Name	External Procedure Names	Method of Operation Diagrams	Program Organization Figures
IDACAT11	Private (User's) Catalog Open	IDACAT11	BB1	—
IDACAT12	Private (User's) Catalog Close	IDACAT12	—	—
IDACAT13	Private Catalog Task Termination	IDACAT13	—	—
IFG0191X	Catalog Open ACB Processor, Load 1	IFG0191X	—	—
IFG0191Y	Catalog Open ACB Processor, Load 2	IFG0191Y	—	—
IFG0200N	VSAM Close/EOV ACB Processor	IFG0200N IFG0550Y	— —	— —
IGC0002F	OS/VS Catalog Management	IGC0002F	AA1, CA1, DB1	8
IGG0CLAA	SUPERLOCATE: Upgrade Support	IGGPSLEN	DG2, DG3, DG4	—
IGG0CLAB	Catalog management driver	IGGPACDV	DB1, DD1, DE1, DH1	8
IGG0CLAC	Open the master catalog	IGGPMCO	—	8
IGG0CLAD	Open the master catalog (part 2) and the catalog recovery area	IGGPMCO2 IGGPRAOP	— —	8 34
IGG0CLAE	DEFINE CATALOG processing: Open and build	IGGPDCME IGGPMEBM	EE2 —	19 —
IGG0CLAF	DELETE CATALOG processing	IGGPDELC IGGPEMIO IGGPEMSG	EB1, EL1 — —	18 — —

Module Directory (continued)

Module Name	Descriptive Name	External Procedure Names	Method of Operation Diagrams	Program Organization Figures
IGG0CLAG	Catalog I/O Subfunctions	IGGPANCI	—	—
		IGGPAOCI	DM2, EF2	10, 18, 19, 24
		IGGPAXCI	—	13, 16, 18, 25
		IGGPCCCR	—	26
		IGGPCPLR	—	—
		IGGPIORA	—	—
		IGGPISCI	—	11, 16, 19, 29
		IGGPPAD	—	10, 18, 22
		IGGPPADC	—	—
		IGGPPDE	—	15, 18, 23
		IGGPPDEC	—	—
		IGGPPUPC	EO1	11, 16, 18, 21
		IGGPRCCR	EO1	18, 27
		IGGPWFLR	—	—
		IGGPXIO	—	28
IGG0CLAH	Search the catalog	IGGPCSDO	—	—
		IGGPQNQC	—	—
		IGGPSCAT	DB1, DC1, EG1	8, 18
		IGGPSC2	BB1, DC1, DC2, DC3	—
IGG0CLAI	DELETE SPACE: Force Support	IGGPDFM1	EK1	18
		IGGPFDSP	—	18
IGG0CLAJ	DEFINE processing: Build data and index catalog records	IGGPDBDI	ED1, ED2	19
IGG0CLAK	DEFINE processing: Catalog record completion	IGGPDCMB	ED2, ED4	19
IGG0CLAL	DEFINE processing: 1st module	IGGPDEF	EC1	18, 19, 38.3
		IGGPDTIM	—	18
IGG0CLAM	SUPER-LOCATE processing	IGGPDBVC	DG5	18
		IGGPSLEL	DG2, DG3	—
		IGGPSLOC	DB1, DG1	8
IGG0CLAN	DEFINE processing: 2nd module	IGGPDCCE	ED1, EE1	—
		IGGPDRDA	EC1, ED1, ED3	—
		IGGPDSCB	EC1	19
		IGGPPSEM	—	—
IGG0CLAO	DEFINE ALIAS Processing	IGGPDEFX	EF2	18
IGG0CLAP	DEFINE processing: 3rd module	IGGPDCDA	EC1, EE1	19
IGG0CLAQ	DEFINE SPACE processing	IGGPDEFS	ED1, ED3, EE1, EG1, EG2	18, 19
IGG0CLAR	Suballocate: Assign candidate volumes	IGGPSALL	DI1, DJ1, ED1, ED3, EE1, EH3	11, 18, 19
IGG0CLAS	DEFINE CATALOG processing	IGGPDCRC	—	—
		IGGPDEFC	EC1, EE1	19
IGG0CLAT	Catalog Management Services Common Processing	IGGPARFS	—	—
		IGGPCDVR	DB1, DD1, EB1, EC1	8, 18, 38.1, 38.3, 38.4, 38.5
IGG0CLAU	Suballocate: Obtain space	IGGPSALS	DJ2	11, 17, 19
IGG0CLAV	Modify a catalog record's field	IGGPDEL2	DM3	10, 15
		IGGPMOD	DM1, DM3, EH3	10, 11, 12, 16, 18, 19
		IGGPSGOP	—	15, 16
		IGGPUPD	DB1, DH1, DI3	8, 10

Module Directory (continued)

Module Name	Descriptive Name	External Procedure Names	Method of Operation Diagrams	Program Organization Figures
IGG0CLAW	Add a group occurrence (set of fields) to a catalog record	IGGPADGO	DM1	10, 13
		IGGPGNEX	—	13, 16
		IGGPGREL	—	13
		IGGPIGOP	DM1	13, 16
		IGGPPREC	DH1	10, 13, 15
IGG0CLAX	Modify a catalog record's field (part 2)	IGGPALT2	DH1, DM2	10, 14
		IGGPDGO	—	15, 16
		IGGPDGOP	—	15, 16
		IGGPEXPD	DM2	14
		IGGPMGO	—	13, 14, 16
		IGGPSHNK	DM2	14
IGG0CLAY	Validate the caller's CTGPL	IGGPSCNC	DB1, DE1, DM1	8, 9, 10
IGG0CLAZ	Extract a catalog record's field	IGGPEXT	DE1, DE2, DJ2	11, 12, 16, 17, 18
		IGGPLOC	DB1, DE1, DE2	8, 9, 12
IGG0CLA0	Show Catalog Processing	IGG0CLA0	EI2	—
IGG0CLA1	Catalog first load	IGG0CLA1	AA1, CA1, DB1	8, 18
IGG0CLA2	Generic LOCATE, Locate GDG Base	IGGPGDGL	DG6	—
		IGGPGLOC	DG5	—
IGG0CLA3	Search the catalog (part 2)	IGGPCSCP	—	—
		IGGPCVX	—	—
		IGGPDOG	—	—
		IGGPGPCC	—	—
		IGGPGRE	DC1, DC3, DC4	—
		IGGPRPLF	DC4, EO1	18
		IGGPRPLM	DC4	18
IGG0CLA4	DEFINE GDGBASE Processing	IGGPDEFB	EB1, EF3	18, 38.7
	DEFINE ALIAS Processing	IGGPDGDS	—	—
		IGGPERUG	—	18
IGG0CLA5	DELETE GDGBASE Processing	IGGPDALL	—	18
		IGGPDELB	EB1, EN1	18
	DELETE ALIAS Processing	IGGPDELX	EB1, EM1	18
IGG0CLA6	DEFINE SPACE: Build the Data Space Group and Data Set Directory Entry sets of fields	IGGPBJFB	EG1	—
		IGGPCBPT	—	—
		IGGPCRTC	EG1	—
		IGGPDFS2	—	—
IGG0CLA7	DELETE processing	IGGPDEMV	—	18
		IGGPDF4T	EJ2	18
		IGGPDUSC	EJ2	18
		IGGPDVMV	—	18
		IGGPMCRA	—	18
		IGGPVMSC	EJ2	18
IGG0CLA8	DEFINE processing	IGGPDFRS	ED2, ED4	—
IGG0CLA9	Catalog Recovery Module	IGGPCCBC	—	—
		IGGPCMRR	—	—

Module Directory (continued)

Module Name	Descriptive Name	External Procedure Names	Method of Operation Diagrams	Program Organization Figures
IGG0CLBA	Extract a catalog record's fields (part 2)	IGGPGREC	DL1, DL2	9, 10, 13
		IGGPGVAL	DE2, DL1 DL2, DM2	9, 10, 14
		IGGPTSTS	DE1, DH1	9, 10
IGG0CLBB	UPDATE-Extend processing	IGGPUPDE	DH1, DI1	10, 11, 16
IGG0CLBC	UPDATE-Extend initialization	IGGPINIT	DI1	11, 16
		IGGPSVOL	DI1	11
| IGG0CLBD	ALTER processing	IGGPALT	EB1, EH1, EH2, EH3	18, 38.2
IGG0CLBE	ALTER: Volume processing	IGGPALEC	—	—
		IGGPALVL	EH3	18
		IGGPANVV	—	—
IGG0CLBF	Subscratch: Return space	IGGPSSCR	DI5, EJ2	12, 18
IGG0CLBG	DELETE Processing	IGGPDEL	EB1, EJ1, EJ4	18
		IGGPDEXA	—	—
		IGGPDLXT	—	—
		IGGPDOPN	EJ1	—
IGG0CLBH	DEFINE NonVSAM Processing	IGGPDAVO	EF1	—
		IGGPDEFA	EB1, EF1	18
IGG0CLBI	Catalog GET and recovery subfunction	IGGPGET	DI4, DI6	8, 9, 10, 11, 12, 13, 15, 16, 18, 19, 20
		IGGPGETC	—	—
		IGGPTNXO	DI5	11, 12, 16
		IGGPTXO	—	11, 12, 16
		IGGPUCRS	—	—
IGG0CLBJ	GENDSP Processing	IGGPGDSP	DB1, DF1	8
IGG0CLBK	LSPACE Processing	IGGPLDCS	DK1	—
		IGGPLSP	DB1, DK1, EG2	8, 18
IGG0CLBL	DELETE SPACE Processing	IGGPDELS	EB1, EK1	18
		IGGPDLET	—	—
		IGGPDLVM	EK1	—
		IGGPDUCB	—	—
| IGG0CLBM	Check the caller's authorization to access catalog information	IGGPCKAU	DB1, DD1	8, 18, 38.6, 38.8
IGG0CLBN	ALTER: Remove volume information	IGGPALMD	—	—
		IGGPALSV	—	—
		IGGPALVR	EH3	18
		IGGPVRD	EH3	18
IGG0CLBO	CRA I/O subfunctions	IGGPRAEA	—	—
		IGGPRAG	—	—
		IGGPRAOR	—	—
		IGGPRAPA	—	31
		IGGPRAPC	—	—
		IGGPRAPD	—	32
		IGGPRAPU	—	30
		IGGPRARC	—	—
		IGGPRASC	—	36
		IGGPRAUR	—	—
		IGGPRAX	—	—
		IGGPXRIO	—	38

Module Directory (continued)

Module Name	Descriptive Name	External Procedure Names	Method of Operation Diagrams	Program Organization Figures
IGG0CLBP	UPDATE-Extend interface to DADSM	IGGPRET1	—	—
		IGGPSPAC	—	11, 16
IGG0CLBQ	LISTCAT Processing	IGGPLSTC	EB1, EI1	18
IGG0CLBR	Bit manipulation routine	IGGPBMR	DJ2	10, 17, 18
IGG0CLBS	Retrieval of derived catalog record information	IGGPXEXT	—	9
		IGGPXVAL	DL1	—
IGG0CLBT	Modification of derived volume catalog record information	IGGPXDGO	DM1	10
		IGGPXEL2	DM3	10
		IGGPXLT2	DM2	10
		IGGPXMOD	DM1	—
IGG0CLBU	Catalog read/write format-4 DSCB	IGGPF4DQ	—	11, 16, 18
		IGGPF4RD	EO1	11, 16, 18
		IGGPF4WR	EO1	11, 16, 18
IGG0CLBV	Catalog SMF record build	IGGPSMFA	ED2, ED4, ED5, EE2	11, 16, 18, 19
		IGGPSMFF	—	—
		IGGPSMFL	DK1	8, 16, 18
		IGGPSMFR	—	—
		IGGPSMFS	—	18
		IGGPSMFZ	—	—
IGG0CLBW	Modify: Delete/Insert Processing	IGGPDEIN	—	14
IGG0CLBX	DEFINE: 4th module	IGGPDCIM	—	—
		IGGPDSPC	ED1, ED3	19
IGG0CLBY	DEFINE: 5th module	IGGPDRSP	ED1, ED3	19
IGG0CLBZ	CONVERTV processing	IGGPCONV	EO1	18
		IGGPGALO	EO1	18, 38.5
		IGGPVALI	EO1	18
IGG0CLB0	Define data and index characteristics	IGGPCMKY	—	—
		IGGPDCCO	—	19
		IGGPDCNV	—	—
		IGGPDPBI	—	—
		IGGPFPDA	—	—
		IGGPITER	—	—
IGG0CLB1	Upgrade management	IGGPUADD	—	18
		IGGPUDEL	EJ3	18
IGG0CLB2	ALTER: Upgrade/Update	IGGPALPG	—	—
		IGGPAUPG	—	18
IGG0CLB3	SMF GET/PUT for alter	IGGPSMF	—	10, 15
		IGGPSMFG	—	9, 10, 13
IGG0CLB4	DEFINE CRA	IGGPDCRA	EE3, EG2	18, 19
		IGGPDCLS	EJ3,EJ4	18
IGG0CLB5	DELETE AIX/path	IGGPDEAX	EJ3	18
		IGGPDEPT	EJ1, EJ3	18
		IGGPDIAX	EJ4	18
		IGGPDIPT	EJ3, EJ4	18
		IGGPDUPG	EJ3, EJ4	18
IGG0CLB6	TSO interface for security	IGGPCKCC	—	—
		IGGPINMD	DD2	—
		IGGPSPSC	DD1	—
		IGGPWTSO	—	8,18

Module Directory (continued)

Module Name	Descriptive Name	External Procedure Names	Method of Operation Diagrams	Program Organization Figures
IGG0CLB7	Reset reusable data set	IGGPRUS	DE1, DI3, DI4, DI5, DI6	10, 12
IGG0CLB8	DEFINE: Space recovery	IGGPCNBO	—	—
		IGGPDFBO	—	19
		IGGPDUND	—	18
IGG0CLB9	VSAM CMS DEFINE	IGGPPAIX	ED3, ED5	19
		IGGPPATH	—	19
IGG0CLCA	CVOL Processor	IGG0CLCA	DB1	—
IGG0CLC9	Catalog request initialization	IGG0CLC9	DB1	8
IGG0CLDA	CRA services	IGGPMODI	EE3	—
		IGGPRAPV	—	—
		IGGPWCAT	EE3	—
		IGGPWCRA	EE3	—
IGG0CKDC	RACF Authorization Check	IGGPRACG	—	38.7
		IGGPRACK	—	38.9
		IGGPRACS	—	—
		IGGPRACT	—	38.6
		IGGPRACV	—	38.8
IGG0CLDD	RACF Definition Handling	IGGPALRI	—	38.1
		IGGPALRN	—	38.2
		IGGPRDEF	—	38.3
		IGGPRDEL	—	18
		IGGPRDLC	—	38.4
		IGGPRFDC	—	18, 38.5
		IGGPROBD	—	—
		IGGPRPTH	—	—
IGG026DU	Catalog Dummy Module	IGG026DU		CA1, DB1

Module Packaging

OS/VS2 Catalog Management modules reside in the pageable link pack area and in the job pack area.

The following table lists the OS/VS2 Catalog Management load modules that are resident in the LPALIB library; they are loaded into the pageable link-pack area by nucleus initialization (NIP) at initial program load (IPL).

Name	Description	OS/VS2 Catalog Management Modules
IGG0CLA1	Catalog management routines	IGG0CLAA, IGG0CLAB, IGG0CLAC, IGG0CLAD, IGG0CLAE, IGG0CLAF, IGG0CLAG, IGG0CLAH, IGG0CLAI, IGG0CLAJ, IGG0CLAK, IGG0CLAL, IGG0CLAM, IGG0CLAN, IGG0CLAO, IGG0CLAP, IGG0CLAQ, IGG0CLAR, IGG0CLAS, IGG0CLAT, IGG0CLAU, IGG0CLAV, IGG0CLAW, IGG0CLAX, IGG0CLAY, IGG0CLAZ, IGG0CLA1, IGG0CLA2, IGG0CLA3, IGG0CLA4, IGG0CLA5, IGG0CLA6, IGG0CLA7, IGG0CLA8, IGG0CLA9, IGG0CLBA, IGG0CLBB, IGG0CLBC, IGG0CLBD, IGG0CLBE, IGG0CLBF, IGG0CLBG, IGG0CLBH, IGG0CLBI, IGG0CLBJ, IGG0CLBK, IGG0CLBL, IGG0CLBM, IGG0CLBN, IGG0CLBO, IGG0CLBP, IGG0CLBQ, IGG0CLBR, IGG0CLBS, IGG0CLBT, IGG0CLBU, IGG0CLBV, IGG0CLBW, IGG0CLBX, IGG0CLBY, IGG0CLBZ, IGG0CLB0, IGG0CLB1, IGG0CLB2, IGG0CLB3, IGG0CLB4, IGG0CLB5, IGG0CLB6, IGG0CLB7, IGG0CLB8, IGG0CLB9, IGG0CLC9, IGG0CLDA, IGG0CLDC, IGG0CLDD
IFG0191X	OS/VS2 Catalog Open ACB Processor, Load 1	IFG0191X
IFG0191Y	OS/VS2 Catalog Open ACB Processor, Load 2	IFG0191Y
IFG0200N	OS/VS2 Catalog Close/End of Volume ACB Processor	IFG0200N

The following module is brought from the LINKLIB into the user's job pack area whenever it is linked to or from the user's program.

Name	Description	OS/VS2 Catalog Management Module
IGG0CLA0	Show catalog processor	IGG0CLA0

External Procedure Name Directory

The external procedure directory is organized alphabetically by symbolic procedure name (external entry point name). It lists the module that contains the procedure, the descriptive name, and the method of operation diagrams and program organization figures in which that procedure is referenced.

Procedure Name	Module Name	Descriptive Name	Method of Operation Diagrams	Program Organization Figures
IDACAT11	IDACAT11	Private (User's) Catalog Open	BB1	—
IDACAT12	IDACAT12	Private (User's) Catalog Close	—	—
IDACAT13	IDACAT13	Private Catalog Task Termination	—	—
IFG0191X	IFG0191X	Catalog Open ACB Processor, Load 1	—	—
IFG0191Y	IFG0191Y	Catalog Open ACB Processor, Load 2	—	—
IFG0200N	IFG0200N	Catalog Close ACB Processor	—	—
IFG0550Y	IFG0200N	Catalog EOV ACB Processor	—	—
IGC0002F	IGC0002F	OS/VS Catalog Management	AA1, CA1, DB1	—
IGGPACDV	IGG0CLAB	Catalog Management Driver	DB1, DD1, DE1, DH1	8
IGGPADGO	IGG0CLAW	Add Group Occurrence (Set of Fields)	DM1	10, 13
IGGPALEC	IGG0CLBE	Check for Index or Data and Sequence Set with Data	—	—
IGGPALMD	IGG0CLBN	Update Security Information	—	—
IGGPALPG	IGG0CLB2	Verify No Open Pagespace	—	—
IGGPALRI	IGG0CLDD	Alter RACF Indicator	—	38.1
IGGPALRN	IGG0CLDD	Rename RACF Profile	—	38.2
IGGPALSV	IGG0CLBN	Sort Volume Serial List	—	—
IGGPALT	IGG0CLBD	ALTER Processing	EB1, EH1, EH2, EH3	18
IGGPALT2	IGG0CLAX	Alter Catalog Record's Field Value	DH1, DM2	10, 14
IGGPALVL	IGG0CLBE	ALTER: Volume Processing	EH3	18
IGGPALVR	IGG0CLBN	ALTER: Volume Processing	EH3	18
IGGPANVV	IGG0CLBE	ALTER: Volume processing	—	—
IGGPAOCI	IGG0CLAG	Assign Contiguous Control Intervals	DM2, EF2	10, 18, 19, 24
IGGPARFS	IGG0CLAT	Tracking Table Spill Routine	—	—
IGGPAUPG	IGG0CLB2	ALTER Upgrade/Update	—	18
IGGPAXCI	IGG0CLAG	Assign One Control Interval	—	13, 16, 18, 25
IGGPBJFB	IGG0CLA6	Build JFCB	EG1	—
IGGPBMR	IGG0CLBR	Bit-Map Manipulation Routine	DJ2	10, 17, 18
IGGPCBPT	IGG0CLA6	Compute Blocks of Track Value	—	—
IGGPCCBC	IGG0CLA9	Catalog Control Block Cleanup	—	—
IGGPCCCR	IGG0CLAG	Checkpoint the Catalog Control Record	—	26
IGGPCDVR	IGG0CLAT	Catalog Management Services: Initial Processing	DB1, DD1, EB1	8, 18
IGGPCKAU	IGG0CLBM	Check the Caller's Authorization To Access the Catalog Record	DB1, DD1	8, 18

External Procedure Name Directory (continued)

Procedure Name	Module Name	Descriptive Name	Method of Operation Diagrams	Program Organization Figures
IGGPCKCC	IGG0CLB6	Check for RACF-Defined Catalog	—	—
IGGPCMKY	IGG0CLB0	Check Key-Range Overlap	—	—
IGGPCMRR	IGG0CLA9	Catalog Management Recovery Routine	—	—
IGGPCNBO	IGG0CLB8	Remove Candidate Volume Occurrences	—	—
IGG026DU		Catalog Dummy Module	CA1, DB1	
IGGPCONV	IGG0CLBZ	CONVERTV Processing	EO1	18
IGGPCPLR	IGG0CLAG	Check Preformat of Low-Key Range	—	—
IGGPCRTC	IGG0CLA6	Convert Records to Tracks	EG1	—
IGGPCSDO	IGG0CLAH	Dynamic Catalog Open Driver	—	—
IGGPCSCP	IGG0CLA3	Search Catalog	—	—
IGGPCVX	IGG0CLA3	CVOL Set-Up Subfunction	—	—
IGGPDALL	IGG0CLA5	Delete Alias Chain	—	18
IGGPDAVO	IGG0CLBH	DEFINE NONVSAM: Build the Volume Information Set of Fields	EF1	—
IGGPDBDI	IGG0CLAJ	DEFINE: Build the Data Set and Index Catalog Records of a Cluster	ED1, ED2	19
IGGPDBVC	IGG0CLAM	DEFINE: Validity-check control blocks	DG5	18
IGGPDCCE	IGG0CLAN	DEFINE: Build the Cluster's Catalog Record	ED1, EE1	—
IGGPDCCO	IGG0CLB0	DEFINE Data and Index Characteristics	—	19
IGGPDCDA	IGG0CLAP	DEFINE CATALOG Processing (2nd of 2)	EC1, EE1	19
IGGPDCIM	IGG0CLBX	Integer Conversion Routine	—	—
IGGPDCLS	IGG0CLB5	DELETE Cluster/AIX Records	EJ3, EJ4	18
IGGPDCMB	IGG0CLAK	DEFINE: Completion (Build the Volume Information Sets of Fields)	ED2, ED4	19
IGGPDCME	IGG0CLAE	DEFINE CATALOG: Catalog Open, Build, and Close	EE2	19
IGGPDCNV	IGG0CLB0	Candidate Volume Determination Routine	ED2	—
IGGPDCRA	IGG0CLB4	DEFINE CRA	EE3, EG2	18, 19
IGGPDCRC	IGG0CLAS	RBA Computing Routine	—	51
IGGPDEAX	IGG0CLB5	Explicit DELETE AIX	EJ3	18
IGGPDEF	IGG0CLAL	DEFINE: Initial Processing	EC1	18, 19
IGGPDEFA	IGG0CLBH	DEFINE NONVSAM Processing	EB1, EF1	18
IGGPDEFB	IGG0CLA4	DEFINE GDGBASE Processing	EB1, EF3	18
IGGPDEFC	IGG0CLAS	DEFINE CATALOG Processing (1st of 2)	EC1, EE1	19
IGGPDEFS	IGG0CLAQ	DEFINE SPACE Processing	ED1, ED3, EE1 EG1, EG2	18, 19
IGGPDEFX	IGG0CLAO	DEFINE ALIAS Processing	EF2	18
IGGPDEIN	IGG0CLBW	Modify: Delete/Insert Processing	—	14
IGGPDEL	IGG0CLBG	DELETE CLUSTER/AIX/PATH/PAGESPACE/ NONVSAM Processing	EB1, EJ1 EJ4	18
IGGPDELB	IGG0CLA5	DELETE GDG BASE processing	EB1, EN1	18

External Procedure Name Directory (continued)

Procedure Name	Module Name	Descriptive Name	Method of Operation Diagrams	Program Organization Figures
IGGPDELC	IGG0CLAF	DELETE CATALOG Processing	EB1, EL1	18
IGGPDELS	IGG0CLBL	DELETE SPACE Processing	EB1, EK1	18
IGGPDELX	IGG0CLA5	DELETE ALIAS NAME processing	EB1, EM1	18
IGGPDEL2	IGG0CLAV	Delete a Set of Fields	DM3	10, 15
IGGPDEMV	IGG0CLA7	DELETE: Extract the volume information set of fields	—	18
IGGPDEPT	IGG0CLB5	Explicit DELETE PATH	EJ1, EJ3	18
IGGPDEXA	IGG0CLBG	Extract Associations	—	—
IGGPDFBO	IGG0CLB8	DEFINE: Space Recovery	—	19
IGGPDFM1	IGG0CLAI	Scratch Format-1 DSCB	EK1	18
IGGPDFRS	IGG0CLA8	Free Unused and Unneeded Storage Resources	ED2, ED4	—
IGGPDFS2	IGG0CLA6	DEFINE SPACE: Build the data space group and data set directory entry sets of fields	—	—
IGGPDF4T	IGG0CLA7	Format-4 Volume Record Timestamp	EJ2	18
IGGPDGDS	IGG0CLA4	Generation Data Set Determination	—	—
IGGPDGO	IGG0CLAX	Modify: Delete Group Occurrence (Set of Fields) Processing	—	15, 16
IGGPDGOP	IGG0CLAX	Modify: Delete Group Occurrence Pointer (Set of Fields Pointer) Processing	—	15, 16
IGGPDIAX	IGG0CLB5	Implicit DELETE AIX	EJ4	18
IGGPDIPT	IGG0CLB5	Implicit DELETE PATH	EJ3, EJ4	18
IGGPDLET	IGG0CLBL	Delete Volume Entry Records	—	—
IGGPDLVM	IGG0CLBL	Build Volume Mount Interface	EK1	—
IGGPDLXT	IGG0CLBG	DELETE Exit Routine	—	—
IGGPDOG	IGG0CLA3	Search Catalog—New Orientation	—	—
IGGPDOPN	IGG0CLBG	OPEN Determination	EJ1	—
IGGPDPBI	IGG0CLB0	Determine Physical Block Size Index Value	—	—
IGGPDRDA	IGG0CLAN	DEFINE AMDSB Processing	EC1, ED1, ED3	—
IGGPDRSP	IGG0CLBY	DEFINE CLUSTER Processing (5th Module)	ED1, ED3	19
IGGPDSCB	IGG0CLAN	DEFINE: Initial Processing (Space Calculations and Build the Cluster Catalog Record)	EC1	19
IGGPDSPC	IGG0CLBX	DEFINE CLUSTER Processing (4th Module)	ED1, ED3	19
IGGPDTIM	IGG0CLAL	DEFINE: Call the system timer	—	18
IGGPDUCB	IGG0CLBL		—	—
IGGPDUND	IGG0CLB8	DEFINE: Undo the Previous Processing	—	18
IGGPDUPG	IGG0CLB5	Upgrade Processing	EJ3, EJ4	18
IGGPDUSC	IGG0CLA7	DELETE: Scratch the data space (format 1—identifier—DSCB) from the volume's VTOC	EJ2	18
IGGPDVMV	IGG0CLA7	DELETE: Mount and verify volumes	—	18
IGGPEMIO	IGG0CLAF	I/O Error Message Writer	—	—

External Procedure Name Directory (continued)

Procedure Name	Module Name	Descriptive Name	Method of Operation Diagrams	Program Organization Figures
IGGPEMSG	IGG0CLAF	Error Message Writer	—	—
IGGPERUG	IGG0CLA4	Delete Reference to Generation Data Set in GDG Base Catalog Record	—	18
IGGPEXPD	IGG0CLAX	Expand a Catalog Record's Variable-length Field	DM2	14
IGGPEXT	IGG0CLAZ	Extract catalog record field	DE1, DE2, DJ2	11, 12 16, 17, 18
IGGPFDSP	IGG0CLAI	Force Delete Space	—	18
IGGPFPDA	IGG0CLB0	Dynamic Allocation Processing	—	—
IGGPF4DQ	IGG0CLBU	Dequeue the Format 4 DSCB	—	11, 16, 18
IGGPF4RD	IGG0CLBU	Read the Format 4 DSCB	EO1	11, 16, 18
IGGPF4WR	IGG0CLBU	Write the Format 4 DSCB	EO1	11, 16, 18
IGGPGALO	IGG0CLBZ	CONVERTV: Get all Low-Range Records	EO1	18
IGGPGDGL	IGG0CLA2	Locate GDG Base catalog record	DG6	—
IGGPGDSP	IGG0CLBJ	LOCATE: GENDSP Processing	DB1, DF1	8
IGGPGET	IGG0CLBI	GET: Call VSAM Record Management to Retrieve A Catalog Record	DI4, DI5	8, 9, 10 11, 12, 13, 15, 16, 18, 19, 20
IGGPGETC	IGG0CLBI	GET: Call VSAM Record Management to Retrieve a Catalog Record	—	—
IGGPGLOC	IGG0CLA2	Generic LOCATE	DG5	—
IGGPGNEX	IGG0CLAW	Format a New Catalog Extension Record	—	13, 16
IGGPGPCC	IGG0CLA3	Obtain PCCB Subfunction	—	—
IGGPGRE	IGG0CLA3	Get real catalog record	DC1, DC3, DC4	—
IGGPGREC	IGG0CLBA	Retrieve a Catalog Record	DL1, DL2	9, 10, 13
IGGPGREL	IGG0CLAW	Get Relative Repetition Number for Add	—	13
IGGPGVAL	IGG0CLBA	Get A Catalog Record Field	DE2, DL1 DL2, DM2	9, 10, 14
IGGPIGOP	IGG0CLAW	Insert A Group Occurrence Pointer (Set-of-Fields Pointer)	DM1	13, 16
IGGPINIT	IGG0CLBC	UPDATE-Extend: Initialization	DI1	11, 16
IGGPINMD	IGG0CLB6	Build User Authorization Interface	DD2	—
IGGPIORA	IGG0CLAG	Decode VSAM Record Management I/O Error Codes	—	—
IGGPISCI	IGG0CLAG	Ensure Catalog Control Interval Availability	—	11, 16, 19, 29
IGGPITER	IGG0CLB0	Adjust Data Control Interval Number Downward	—	—
IGGPLDCS	IGG0CLBK	LSPACE: Gather the Available-Space Data	DK1	—
IGGPLOC	IGG0CLAZ	Locate catalog data fields	DB1, DE1, DE2	8, 9, 12
IGGPLSP	IGG0CLBK	LSPACE Processing	DB1, DK1, EG2,	8,18
IGGPLSTC	IGG0CLBQ	LISTCAT Processing	EB1, EI1	18

External Procedure Name Directory (continued)

Procedure Name	Module Name	Descriptive Name	Method of Operation Diagrams	Program Organization Figures
IGGPMCO	IGG0CLAC	DEFINE CATALOG: Master Catalog Build and Open (1st of 2)	—	8
IGGPMCO2	IGG0CLAD	DEFINE CATALOG: Master Catalog Build and Open (2nd of 2)	—	8
IGGPMCRA	IGG0CLA7	DELETE: Verify CRA Volume Mount	—	18
IGGPMEBM	IGG0CLAE	Handle multiple extents for catalog open and build	—	—
IGGPMGO	IGG0CLAX	Move Group Occurrences (Sets-of-Fields) From One Extension Record Into Another	—	13, 14, 16
IGGPMOD	IGG0CLAV	Modify: Initial Processing	DM1, DM3, EH3	10, 11, 12, 16, 18, 19
IGGPMODI	IGG0CLDA	Build Interface to MODIFY Function	EE3	—
IGGPPAD	IGG0CLAG	PUT Add: Call VSAM Record Management To Write A New Catalog Record	—	10, 18, 22
IGGPPADC	IGG0CLAG	—	—	—
IGGPPAIX	IGG0CLB9	DEFINE AIX	ED3, ED5	19
IGGPPATH	IGG0CLB9	DEFINE PATH	—	19
IGGPPDE	IGG0CLAG	ERASE: Call VSAM Record Management to Erase A Catalog Record	—	15, 18, 23
IGGPPDEC	IGG0CLAG	ERASE: Call VSAM Record Management to Erase a Catalog Control Interval	—	18
IGGPPREC	IGG0CLAW	Call PUT-Add or PUT-Update to Write a Catalog Record, Then Call SMF	DH1	10, 13, 15
IGGPPSEM	IGG0CLAN	DEFINE: Validity-check Input Parameters	—	—
IGGPPUPC	IGG0CLAG	PUT-Update: Call VSAM Record Management to Rewrite A Catalog Record	EO1	11, 16, 18, 21
IGGPQNQC	IGGPCLAH	Check for entryname conflicts	—	—
IGGPRACC	IGG0CLBO	Checkpoint CRA CCR	—	—
IGGPRACG	IGG0CLDC	RACHECK for Implicit Alter Delete	—	38.7
IGGPRACK	IGG0CLDC	RACHECK Routine	—	38.9
IGGPRACS	IGG0CLDC	Perform Set-Up for Extract	—	—
IGGPRACT	IGG0CLDC	RACF Catalog Verification	—	38.6
IGGPRACV	IGG0CLDC	RACF Verification Routine	—	38.8
IGGPRAEA	IGG0CLBO	CRA I/O Error Analysis	—	—
IGGPRAG	IGG0CLBO	CRA GET	—	—
IGGPRAOP	IGG0CLAD	Open CRA	—	34
IGGPRAOR	IGG0CLBO	Orient to CRA	—	—
IGGPRAPA	IGG0CLBO	CRA Put Add	—	31
IGGPRAPC	IGG0CLBO	CRA Record Chain	—	—
IGGPRAPD	IGG0CLBO	CRA Put Delete	—	32
IGGPRAPU	IGG0CLBO	CRA Put Update	—	30
IGGPRAPV	IGG0CLDA	Format-4 DSCB Timestamp Processing	—	—

External Procedure Name Directory (continued)

Procedure Name	Module Name	Descriptive Name	Method of Operation Diagrams	Program Organization Figures
IGGPRARA	IGG0CLBO	Read CRA CCR	—	35
IGGPRASC	IGG0CLBO	Ensure Sufficient CRA Control Intervals	—	36
IGGPRAX	IGG0CLBO	CRA Exit Routine	—	37
IGGPRCCR	IGGPCLAG	Read the catalog control record (CCR)	EO1	18, 27
IGGPRDEF	IGG0CLDD	Define a RACF Profile	—	38.3
IGGPRDEL	IGG0CLDD	Delete a RACF Profile	—	18
IGGPRDLC	IGG0CLDD	Delete RACF Profiles for Partially-Defined Cluster	—	38.4
IGGPRET1	IGG0CLBP	Return point from DADSM EXTEND	—	—
IGGPRFDC	IGG0CLDD	RACF Force Delete Catalog	—	38.5
IGGPROBD	IGG0CLDD	RACF Delete Array of Profiles	—	—
IGGPRPLF	IGG0CLA3	Drop catalog orientation for this request	DC4, EO1	18
IGGPRPLM	IGG0CLA3	RPL Manager	DC4	18
IGGPRPTH	IGG0CLDD	RACF Delete Profile for Path	—	—
IGGPRUS	IGG0CLB7	Reset Reusable Data Set	DE1, DI3, DI4, DI5, DI6	10, 12
IGGPSALL	IGG0CLAR	Suballocate: Candidate Volume Assignment	DI1, DJ1, ED1, ED3, EE1, EM3	11, 18, 19
IGGPSALS	IGG0CLAU	Suballocate: Space Assignment	DJ2	11, 17, 19
IGGPSCAT	IGG0CLAH	Search Catalog Processing	DB1, DC1, EG1	8, 18
IGGPSCNC	IGG0CLAY	Initial CTGPL Processing	DB1, DE1, DM1	8, 9, 10
IGGPSC2	IGG0CLAH	Search the catalog (part 2)	BB1, DC1, DC2, DG5	—
IGGPSGOP	IGG0CLAV	Search For a Group Occurrence Pointer (Set-of-Fields Pointer)	—	15, 16
IGGPSHNK	IGG0CLAX	Shrink A Catalog Record's Variable-Length Field	DM2	14
IGGPSLEI	IGG0CLAM	Extract Initialization: Build CTGPL, CTGFLs	—	—
IGGPSLEL	IGG0CLAM	Process Elementary Entry	DG2, DG3	—
IGGPSLEN	IGG0CLAA	Process Base Data Record	DG2, DG3, DG4	—
IGGPSLOC	IGG0CLAM	LOCATE: Super-Locate Processing	DB1, DG1	8
IGGPSMF	IGG0CLB3	SMF-for-ALTER: Write Processing	—	10, 15
IGGPSMFA	IGG0CLBV	SMF: Build a Format-63 SMF Record	ED2, ED4, ED5, EE2	19
IGGPSMFF	IGG0CLBV	SMF: Free the Virtual Storage Obtained	—	—
IGGPSMFG	IGG0CLB3	SMF-for-ALTER: Read Processing	—	9, 10, 13
IGGPSMFL	IGG0CLBV	SMF: Build a Format-69 SMF Record	DK1	8, 16, 18
IGGPSMFR	IGG0CLBV	SMF: Build a Format-68 SMF Record	—	—
IGGPSMFS	IGG0CLBV	SMF: Build a Format-67 SMF Record	—	18
IGGPSMFZ	IGG0CLBV	SMF: Set the Password Set of Fields to Zero	—	—
IGGPSPAC	IGG0CLBP	UPDATE-Extend: Obtain More Space	—	11, 16
IGGPSPSC	IGG0CLB6	Check for Security By-Pass	DD1	—

External Procedure Name Directory (continued)

Procedure Name	Module Name	Descriptive Name	Method of Operation Diagrams	Program Organization Figures
IGGPSSCR	IGG0CLBF	Subscratch: Release A Cluster's Space Within a VSAM Data Space		
IGGPSVOL	IGG0CLBC	Search For the Volume Information Set of Fields	DI1	11
IGGPTNXO	IGG0CLBI	Calculate SUMTT Value	DI5	11, 12, 16
IGGPTSTS	IGG0CLBA	CTGFL-for-Tests Processing	DE1, DH1	9, 10
IGGPTXO	IGG0CLBI	Update SUMTT in Volume Record	—	11, 12, 16
IGGPUADD	IGG0CLB1	Add Upgrade Association	—	18
IGGPUCRS	IGG0CLBI	GET a CRA Record by CI Number	—	—
IGGPUDEL	IGG0CLB1	Delete Upgrade Association	EJ3	18
IGGPUPD	IGG0CLAV	UPDATE: Initial Processing	DB1, DH1,DI3	8, 10
IGGPUPDE	IGG0CLBB	UPDATE-Extend Processing	DH1, DI1	10, 11, 16
IGGPVALI	IGG0CLBZ	CONVERTV: Validity Checking	EO1	18
IGGPVRD	IGG0CLBN	ALTER: Remove VSAM Space from Volume	EH3	18
IGGPVMSC	IGG0CLA7	DELETE: Delete all space information in the the volume catalog record	EJ2	18
IGGPWCAT	IGG0CLDA	Write Volume Records	EE3	—
IGGPWCRA	IGG0CLDA	Write Self-Describing and Volume Records to CRA	EE3	—
IGGPWFLR	IGG0CLAG	Write Free Record	—	—
IGGPWTSO	IGG0CLB6	Write Messages To A TSO Terminal User	—	8, 18
IGGPXDGO	IGG0CLBT	Add Derived Group Occurrence	DM1	10
IGGPXEL2	IGG0CLBT	Delete Derived Group Occurrence	DM3	10
IGGPXEXT	IGG0CLBS	Extract Derived Group Occurrence	—	9
IGGPXIO	IGG0CLAG	Issue the GET, PUT, or ERASE Macro Instructions to VSAM Record Management	—	28
IGGPXLT2	IGG0CLBT	Alter Derived Field Value	DM2	10
IGGPXMOD	IGG0CLBT	Modify Derived Group Occurrence	DM1	—
IGGPXRIO	IGG0CLBO	CRA GET/PUT Routine	—	38
IGGPXVAL	IGG0CLBS	Get Derived Field Value	DL1	—
IGG0CLA0	IGG0CLA0	Show Catalog Processing	EI2	—
IGG0CLA1	IGG0CLA1	Catalog Management First Load	AA1, CA1, DB1	8,18
IGG0CLCA	IGG0CLCA	CVOL Processor	DB1	—
IGG0CLC9	IGG0CLC9	Catalog Management Initialization	DB1	8

Procedure Calls Directory

This table contains each Catalog Management module and external and internal procedures within each module. The internal and external calls of each procedure are listed.

Calling Module and its procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLAA		
External		
IGGPSLEN	IGGPSLEL (AM)	IGGPSLCG, IGGPSLIV, IGGPSLR, IGGPSLY
Internal		
IGGPSLCG	IGGPEXT (AZ), IGGPGET (BI)	IGGPSLY
IGGPSLIV	—	—
IGGPSLR	IGGPEXT (AZ), IGGPGET (BI), IGGPSLEL (AM)	IGGPSLY
IGGPSLY	IGGPEXT (AZ), IGGPGET (BI), IGGPSLEL (AM)	—
IGG0CLAB		
External		
IGGPACDV	IGGPCDVR (AT), IGGPCKAU (BM), IGGPGDSP (BJ), IGGPLOC (AZ), IGGPLSP (BK), IGGPMCO (AC), IGGPSCAT (AH), IGGPSCNC (AY), IGGPSLOC (AM), IGGPUPD (AV)	—
IGG0CLAC		
External		
IGGPMCO	IGGPMCO2 (AD)	—
IGG0CLAD		
External		
IGGPMCO2	—	—
IGGPRAOP	—	—
IGG0CLAE		
External		
IGGPDCME	IGGPDCRC (AS), IGGPMOD (AV)	IGGPDCOC
IGGPMEBM	IGGPBMR (BR)	—
Internal		
IGGPDCBO	IGGPDUND (B8), IGGPF4RD (BU), IGGPF4WR (BU)	—
IGGPDCCB	—	—
IGGPDCOC	IGGPARFS (AT), IGGPMCO2 (AD), IGGPSMFA (BV)	IGGPDCBO, IGGPDCCB, IGGPDCPR
IGGPDCPR	IGGPXIO (AG)	—

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLAF		
External		
IGGPDELC	IGGPCCCR (AG), IGGPDALL (A5), IGGPDBVC (AM), IGGPEXT (AZ), IGGPFDSP (AI), IGGPF4RD (BU), IGGPF4WR (BU), IGGPGET (BI), IGGPPDE (AG), IGGPRPLF (A3), IGGPSCAT (AH), IGGPXIO (AG)	IGGPDLER, IGGPSDSP
IGGPEMIO	—	—
IGGPEMSG	—	IGGPNFND
Internal		
IGGPDCDS	IGGPARFS (AT), IGGPDOPN (BG), IGGPEXT (AZ)	—
IGGPDLER	IGGPDF4T (A7)	—
IGGPNFND	—	—
IGGPSDSP	—	—
IGG0CLAG		
External		
IGGPANCI	—	IGGPCHAC, IGGPIORA, IGGPWFLR, IGGPXIO
IGGPAOCI	—	IGGPANCI, IGGPCCCR, IGGPRCCR, IGGPXIO
IGGPAXCI	—	IGGPANCI, IGGPCCCR, IGGPRCCR, IGGPXIO
IGGPCCCR	—	IGGPRBAP, IGGPXIO
IGGPIORA	IGGPEMIO (AF), IGGPEMSG (AF)	—
IGGPISCI	IGGPGET (BI), IGGPRASC (BO)	IGGPANCI, IGGPCCCR, IGGPRCCR, IGGPXIO
IGGPPAD	IGGPRAPA (BO), IGGPRAPU (BO)	IGGPRCCR, IGGPXIO
IGGPPADC	—	—
IGGPPDE	IGGPRAPD (BO)	IGGPRCCR, IGGPTRPL, IGGPXIO
IGGPPDEC	—	—
IGGPPUPC	IGGPRAPU (BO)	IGGPTRPL, IGGPXIO
IGGPRCCR	—	IGGPRBAP, IGGPXIO
IGGPXIO	—	IGGPIORA
Internal		
IGGPCHAC	—	—
IGGPCPLR	—	IGGPWFLR
IGGPRBAP	—	—
IGGPTRPL	—	IGGPXIO
IGGPWFLR	—	IGGPIORA, IGGPXIO

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLAH		
External		
IGGPCSDO	IGGPCVX (A3), IGGPDOG (A3)	IGGPACI
IGGPQNQC	IGGPGET (BI)	—
IGGPSCAT	IGGPUCRS (BI)	IGGPIOSI, IGGPQNQC, IGGPSCA, IGGPSC2
IGGPSC2	IGGPSCCP (A3), IGGPDOG (A3)	IGGPPCCP
Internal		
IGGPACI	IGGPGPCC (A3), IGGPRPLF (A3)	—
IGGPIOSI	—	—
IGGPPCCP	IGGPDOG (A3), IGGPGPCC (A3)	IGGPCSDO
IGGPSCA	—	—
IGG0CLAI		
External		
IGGPDFM1	IGGPF4DQ (BU), IGGPF4RD (BU)	—
IGGPFDSP	IGGPARFS (AT), IGGPDLET (BL), IGGPDLVM (BL), IGGPDOPN (BG), IGGPDUCB (BL), IGGPEXT (AZ), IGGPGET (BI), IGGPMOD (AV), IGGPPUPC (AG)	IGGPFDEX, IGGPFDXT
Internal		
IGGPFDEX	—	—
IGGPFDXT	—	—
IGG0CLAJ		
External		
IGGPDBDI	IGGPDCMB (AK), IGGPDFBO (B8), IGGPDFRS (A8), IGGPDTIM (AL), IGGPGET (BI), IGGPSMFA (BV), IGGPUADD (B1)	IGGPDCNV, IGGPDRCS, IGGPDSPO
Internal		
IGGPDCNV	IGGPARFS (AT), IGGPSALL (AR) IGGPDCNV (B0)	—
IGGPDEXD	IGGPARFS (AT)	—
IGGPDRCS	—	—
IGGPDSEX	IGGPARFS (AT)	IGGPDEXD
IGGPDSPO	IGGPCNBO (B8), IGGPDEFS (AQ), IGGPGET (BI), IGGPSALL (AR), IGGPTNXO (BI), IGGPTXO (BI)	IGGPDSEX
IGG0CLAK		
External		
IGGPDCMB	—	IGGPDBVO, IGGPDMOP, IGGPDOMF

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPDBCV	IGGPARFS (AT)	IGGPDEXD
IGGPDBVO	IGGPARFS (AT)	IGGPDBCV, IGGPDEXD, IGGPDRCA, IGGPDRNG, IGGPDSSP
IGGPDEXD	IGGPARFS (AT)	—
IGGPDMOP	IGGPARFS (AT), IGGPMOD (AV)	—
IGGPDOMF	IGGPSMFA (BV)	—
IGGPDRCA	—	—
IGGPDRNG	—	—
IGGPDSSP	—	IGGPDEXD
IGG0CLAL		
External		
IGGPDEF	IGGPDSCB (AN)	IGGPDCAV, IGGPDDEP
IGGPDTIM	—	—
Internal		
IGGPDBVC	—	—
IGGPDCAV	—	IGGPDBVC, IGGPDCSF, IGGPDCWC, IGGPDDNP, IGGPDFSC, IGGPDRPG
IGGPDCDE	—	—
IGGPDCSF	—	IGGPDBVC
IGGPDCWC	—	IGGPDBVC
IGGPDDEP	IGGPFPDA (B0)	IGGPDCDE, IGGPDEDE, IGGPDSTY, IGGPDWAI
IGGPDDNP	—	—
IGGPDEDE	IGGPGET (BI), IGGPQNQC (AH), IGGPRDEF (DD)	IGGPDTIM
IGGPDFSC	—	IGGPDBVC
IGGPDRPG	IGGPCMKY (B0)	—
IGGPDSTY	—	IGGPDBVC
IGGPDWAI	—	—
IGG0CLAM		
External		
IGGPDBVC	—	—
IGGPSLEL	IGGPEXT (AZ)	—
IGGPSLOC	IGGPGDGL (A2), IGGPGLOC (A2), IGGPSLEN (AA)	IGGPSLIN

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPSLEI	IGGPEXT (AZ), IGGPGET (BI)	IGGPSLEX
IGGPSLER	IGGPEXT (AZ)	—
IGGPSLEX	—	—
IGGPSLE2	—	—
IGGPSLIN	—	IGGPDBVC, IGGPSLEL, IGGPSLE2
IGG0CLAN		
External		
IGGPDCCE	IGGPAOCI (AG), IGGPPAD (AG)	—
IGGPDRDA	IGGPDRSP (BY), IGGPDSPC (BX)	IGGPDCCE
IGGPDSCB	IGGPDBDI (AJ), IGGPDCDA (AP), IGGPDEFC (AS)	IGGPDALR, IGGPDBSF, IGGPDRDA, IGGPDSPF, IGGPDUND
IGGPPSEM	—	IGGPXDCI
Internal		
IGGPDALR	—	—
IGGPDBSF	—	—
IGGPDSPF	—	—
IGGPDUND	—	—
IGGPXDCI	—	—
IGG0CLAO		
External		
IGGPDEFX	IGGPAXCI (AG), IGGPEXT (AZ), IGGPGET (BI), IGGPMOD (AV), IGGPPDE (AG), IGGPSMFA (BV)	IGGPBCFI, IGGPMCFI
Internal		
IGGPBCFI	IGGPMOD (AV)	—
IGGPMCFI	—	—
IGG0CLAP		
External		
IGGPDCDA	—	IGGPDCVS
Internal		
IGGPDCON	—	—
IGGPDCPC	IGGPAXCI (AG), IGGPDCCE (AN), IGGPPAD (AG)	—
IGGPDCSP	—	IGGPDCON, IGGPDCPC
IGGPDCVS	—	—

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLAQ		
External		
IGGPDEFS	IGGPARFS (AT), IGGPAXCI (AG), IGGPBJFB (A6), IGGPCBPT (A6), IGGPCKAU (BM), IGGPCRTC (A6), IGGPDCRA (B4), IGGPDFS2 (A6), IGGPDTIM (AL), IGGPGET (BI), IGGPISCI (AG), IGGPLSP (BK), IGGPPAD (AG), IGGPSCAT (AH)	IGGPCOBT, IGGPDSXT, IGGPF1BO, IGGPF4PR, IGGPIVER, IGGPVMTV
Internal		
IGGPCOBT	—	—
IGGPDSXT	IGGPARFS (AT)	—
IGGPF1BO	—	—
IGGPF4PR	IGGPF4DQ (BU), IGGPF4RD (BU), IGGPF4WR (BU)	—
IGGPIVER	—	—
IGGPVMTV	—	—
IGG0CLAR		
External		
IGGPSALL	IGGPEXT (AZ), IGGPGET (BI), IGGPISCI (AG), IGGPMOD (AV), IGGPSALS (AU)	—
IGG0CLAS		
External		
IGGPDCRC	—	—
IGGPDEFC	IGGPCCCR (AG), IGGPDCME (AE)	IGGPDCBE, IGGPDCSP, IGGPDCVO
Internal		
IGGPDCBE	—	IGGPDCEB, IGGPDCPB
IGGPDCEB	—	IGGPDCME
IGGPDCFL	—	—
IGGPDCHD	—	IGGPDCRC
IGGPDCIX	—	IGGPDCRC
IGGPDCLD	—	IGGPDCRC
IGGPDCME	—	—
IGGPDCPB	—	—
IGGPDCSP	IGGPARFS (AT), IGGPDCRA (B4), IGGPDEFS (AQ), IGGPMEBM (AE), IGGPSALL (AR)	—
IGGPDCVO	—	IGGPDCFL, IGGPDCHD, IGGPDCIX, IGGPDCLD

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLAT		
External		
IGGPARFS	—	—
IGGPCDVR	IGGPALRI (DD), IGGPALT (BD), IGGPCKAU (BM), IGGPCONV (BZ), IGGPDEF (AL), IGGPDEFA (BH), IGGPDEFB (A4), IGGPDEFS (AQ), IGGPDEFX (A4), IGGPDEL (BG), IGGPDELB (A5), IGGPDELC (AF), IGGPDELS (BL), IGGPDELX (A5), IGGPGET (BI), IGGPLSTC (BQ), IGGPRDEF (DD), IGGPRDEL (DD), IGGPRDLC (DD), IGGPRFDC (DD), IGGPSCAT (AH)	IGGPARFS, IGGPCCLN, IGGPDYNA, IGGPDYNB, IGGPDYNC, IGGPDYND
Internal		
IGGPCCLN	—	IGGPARFS
IGGPDYNA	IGGPRPLF (A3), IGGPRPLM (A3)	IGGPARFS, IGGPDYNB, IGGPDYND
IGGPDYNB	—	IGGPDYNC
IGGPDYNC	—	—
IGGPDYND	IKJEFF18 (IKJEFF18)	—
IGG0CLAU		
External		
IGGPSALS	IGGPBMR (BR), IGGPEXT (AZ)	IGGPEDS
Internal		
IGGPEDS	IGGPBMR (BR)	—
IGG0CLAV		
External		
IGGPDEL2	IGGPDGO (AX), IGGPDGOP (AX), IGGPGREC (BA), IGGPPDE (AG), IGGPPREC (AW), IGGPSMF (B3)	IGGPSGOP
IGGPMOD	IGGPSCNC (AY), IGGPXMOD (BT)	IGGPSFPL
IGGPSGOP	IGGPGREC (BA)	—
IGGPUPD	IGGPRUS (B7), IGGPUPDE (BB)	IGGPSFPL
Internal		
IGGPSFPL	IGGPGREC (BA), IGGPPREC (AW), IGGPSMFG (B3), IGGPTSTS (BA), IGGPXDGO (BT), IGGPXEL2 (BT), IGGPXLT2 (BT)	—

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLAW		
External		
IGGPADGO	IGGPAXCI (AG), IGGPGREC (BA)	IGGPAGOP, IGGPASPT, IGGPGNEX, IGGPGREL, IGGPIGOP, IGGPMVGO
IGGPGNEX	—	IGGPPREC
IGGPGREL	IGGPGREC (BA)	—
IGGPIGOP	—	—
IGGPPREC	IGGPPAD (AG), IGGPPUPC (AG), IGGPSMF (B3)	—
Internal		
IGGPAGOP	IGGPAXCI (AG), IGGPGREC (BA), IGGPMGO (AX)	IGGPASPT, IGGPGNEX, IGGPGREL, IGGPIGOP
IGGPASPT	—	IGGPIGOP
IGGPMVGO	—	—
IGG0CLAX		
External		
IGGPALT2	IGGPAOCI (AG), IGGPGVAL (BA)	IGGPMVAR
IGGPDGO	—	—
IGGPDGOP	—	—
IGGPEXPD	—	—
IGGPMGO	IGGPIGOP (AW), IGGPSGOP (AV)	IGGPCGO, IGGPDGO, IGGPDGOP
IGGPSHNK	—	—
Internal		
IGGPCGO	—	—
IGGPMBGO	IGGPAXCI (AG), IGGPGNEX (AW), IGGPGREC (BA)	IGGPMGO
IGGPMVAR	IGGPAXCI (AG), IGGPDEIN (BW), IGGPGNEX (AW), IGGPGREC (BA), IGGPGVAL (BA)	IGGPEXPD, IGGPMBGO, IGGPMGO, IGGPSHNK
IGG0CLAY		
External		
IGGPSCNC	—	IGGPSNVC
Internal		
IGGPSNVC	—	—
IGG0CLAZ		
External		
IGGPEXT	IGGPSCNC (AY), IGGPXEXT (BS)	IGGPSCNF
IGGPLOC	—	IGGPSCNF

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPGREP	IGGPGREC (BA)	—
IGGPLOC2	IGGPGVAL (BA)	IGGPGREP, IGGPSHIN
IGGPSCNF	IGGPGREC (BA), IGGPSMFG (B3), IGGPTSTS (BA)	IGGPLOC2, IGGPUPGD
IGGPSHIN	—	—
IGGPUPGD	IGGPGET (BI), IGGPGVAL (BA)	—
IGG0CLA0		
External		
IGG0CLA0	—	—
IGG0CLA1		
External		
IGG0CLA1	IGG0CLC9	—
IGG0CLA2		
External		
IGGPGDGL	—	IGGPGFND, IGGPGGE, IGGPGMIN, IGGPGRGN, IGGPGTAB
IGGPGLOC	IGGPDBVC (AM), IGGPGET (BI), IGGPSC2 (AH)	IGGPGLCN
Internal		
IGGPGFND	—	—
IGGPGGE	IGGPEXT (AZ), IGGPGET (BI)	—
IGGPGLCN	—	—
IGGPGMIN	—	—
IGGPGRGN	—	—
IGGPGTAB	IGGPEXT (AZ), IGGPSLEL (AM)	—
IGG0CLA3		
External		
IGGPCSCP	IGGPCSDO (AH)	IGGPCPCC, IGGPDOG, IGGPGPCC
IGGPCVX	—	—
IGGPDOG	IGGPARFS (AT), IGGPCKAU (BM), IGGPGET (BI)	IGGPCVX, IGGPGRE, IGGPRPLF, IGGPRPLM
IGGPGPCC	—	—
IGGPGRE	IGGPGET (BI)	—
IGGPRPLF	—	—
IGGPRPLM	IGGPCCBC (A9)	IGGPSRPL
Internal		
IGGPCPCC	—	—
IGGPSRPL	—	—

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLA4		
External		
IGGPDEFB	IGGPAXCI (AG), IGGPPAD (AG), IGGPSMFA (BV)	—
IGGPDGDS	IGGPARFS (AT), IGGPDEL (BG), IGGPEXT (AZ), IGGPGET (BI), IGGPMOD (AV), IGGPPUPC (AG) IGGPRACG (DC)	IGGPBCFA, IGGPBCFC, IGGPBCFD, IGGPGCOM, IGGPGEXP
IGGPERUG	IGGPEXT (AZ), IGGPMOD (AV)	IGGPBCFC, IGGPBCFD, IGGPGCOM
Internal		
IGGPBCFA	—	IGGPCCFB
IGGPBCFC	—	IGGPCCFB
IGGPBCFD	—	—
IGGPCCFD	—	—
IGGPGCOM	—	—
IGGPGEXP	—	—
IGG0CLA5		
External		
IGGPDALL	IGGPEXT (AZ), IGGPGET (BI)	IGGPCCFM, IGGPDONE
IGGPDELB	IGGPARFS (AT), IGGPDEL (BG), IGGPEXT (AZ), IGGPGET (BI), IGGPMOD (AV), IGGPPDE (AG), IGGPSMFS (BV)	IGGPBCFE, IGGPDGBC, IGGPPUNT
IGGPDELX	IGGPGET (BI)	IGGPCCFM, IGGPDONE
Internal		
IGGPBCFE	—	—
IGGPCCFM	—	—
IGGPDGBC	IGGPARFS (AT)	—
IGGPDONE	IGGPEXT (AZ), IGGPGET (BI), IGGPMOD (AV), IGGPPDE (AG), IGGPSMFS (BV)	IGGPCCFM
IGGPPUNT	—	—
IGG0CLA6		
External		
IGGPBJFB	IGGPARFS (AT)	—
IGGPCBPT	—	—
IGGPCRTC	—	—
IGGPDFS2	—	IGGPCDSD, IGGPCSDG, IGGPCSHG

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPCDSD	IGGPSALL (AR)	IGGPDSMD
IGGPCSDG	—	IGGPDSMD
IGGPCSHG	—	IGGPDSMD
IGGPDSMD	IGGPMOD (AV)	
IGG0CLA7		
External		
IGGPDEMV	IGGPEXT (AZ)	IGGPSET
IGGPDF4T	IGGPDLXT (BG), IGGPF4DQ (BU), IGGPF4RD (BU), IGGPF4WR (BU), IGGPPUPC (AG)	
IGGPDUSC	IGGPDLXT (BG)	—
IGGPDVMV	—	—
IGGPMCRA	—	IGGPDVMV
IGGPVMSC	IGGPGET (BI), IGGPSSCR (BF), IGGPARFS (AT)	IGGPDEDD, IGGPDMSG, IGGPDESH, IGGPDEVG, IGGPDUSC
Internal		
IGGPDEDD	IGGPMOD (AV)	IGGPSET
IGGPDESH	IGGPMOD (AV)	IGGPSET
IGGPDEVG	IGGPARFS (AT), IGGPEXT (AZ), IGGPDLXT (BG), IGGPMOD (AV)	IGGPDVMV, IGGPSET
IGGPDMSG	IGGPEXT (AZ), IGGPMOD (AV)	IGGPRCTV, IGGPSET
IGGPRCTV	IGGPF4DQ (BU), IGGPF4RD (BU), IGGPF4WR (BU)	—
IGGPSET	—	—
IGGPRCTV	—	—
IGG0CLA8		
External		
IGGPDFRS	IGGPARFS (AT)	—
Internal		
IGGPDFRE	IGGPARFS (AT)	—
IGG0CLA9		
External		
IGGPCCBC	—	—
IGGPCMRR	IGGPCCCR (AG), IGGPRPLF (A3)	IGGPCRPL, IGGPDDBO, IGGPFRES, IGGPPDO

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPCRPL	—	IGGPCCBC
IGGPDDBO	IGGPCCCR (AG), IGGPCDVR (AT), IGGPRPLF (A3)	—
IGGPFRES	—	IGGPGNTE
IGGPGNTE	—	—
IGGPPDO	—	—
IGG0CLBA		
External		
IGGPGREC	IGGPGET (BI), IGGPPREC (AW), IGGPSMFG (B3)	—
IGGPGVAL	IGGPXVAL (BS)	IGGPGREC, IGGPLVAL
IGGPTSTS	—	IGGPGVAL, IGGPTCMP, IGGPGREC
Internal		
IGGPCKLC	—	—
IGGPLVAL	—	IGGPCKLC
IGGPTCMP	—	—
IGG0CLBB		
External		
IGGPUPDE	IGGPEXT (AZ), IGGPGET (BI), IGGPINIT (BC), IGGPSMFA (BV), IGGPSMFL (BV), IGGPSVOL (BC), IGGPTNXO (BI), IGGPTXO (BI)	IGGPCEXT, IGGPCSAL, IGGPMEXT, IGGPMVOL, IGGPSSWD, IGGPUALL
Internal		
IGGPCEXT	—	—
IGGPCSAL	IGGPARFS (AT), IGGPSALL (AR), IGGPSPAC (BP)	—
IGGPMEXT	IGGPMOD (AV)	—
IGGPMVOL	IGGPMOD (AV)	—
IGGPSSWD	IGGPGET (BI), IGGPINIT (BC), IGGPSVOL (BC)	IGGPCEXT, IGGPMEXT, IGGPMVOL
IGGPUALL	IGGPGET (BI)	
IGG0CLBC		
External		
IGGPINIT	IGGPEXT (AZ)	—
IGGPSVOL	IGGPEXT (AZ)	—
IGG0CLBD		
External		
IGGPALT	IGGPALMD (BN), IGGPALVL (BE), IGGPANVV (BE), IGGPARFS (AT), IGGPAUPG (B2), IGGPGET (BI), IGGPPUPC (AG), IGGPSMF (B3), IGGPSMFA (BV), IGGPSMFG (B3)	IGGPALNM

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPALF1	IGGPARFS (AT)	IGGPALGV, IGGPALSV
IGGPALGV	IGGPARFS (AT), IGGPEXT (AZ)	—
IGGPALNM	IGGPALRN (DD), IGGPGET (BI), IGGPPAD (AG), IGGPPDE (AG), IGGPQNQC (AH), IGGPSMFR (BV)	IGGPALF1
IGGPALSV	IGGPARFS (AT)	—
IGG0CLBE		
External		
IGGPALEC	IGGPGET (BI)	IGGPALIX
IGGPALVL	IGGPALVR (BN), IGGPARFS (AT), IGGPEXT (AZ), IGGPVRD (BN)	IGGPALBC, IGGPALVA
IGGPANVV	IGGPDAVO (BH), IGGPMOD (AV)	—
Internal		
IGGPALAE	—	—
IGGPALBC	—	—
IGGPALIX	IGGPEXT (AZ), IGGPGET (BI)	IGGPALBC
IGGPALSA	IGGPMOD (AV), IGGPSALL (AR)	IGGPALBC, IGGPALEC
IGGPALVA	IGGPEXT (AZ)	IGGPALBC, IGGPALAE, IGGPALSA
IGG0CLBF		
External		
IGGPSSCR	IGGPBMR (BR)	—
IGG0CLBG		
External		
IGGPDEL	IGGPDALL (A6), IGGPDCLS (B5), IGGPDEAX (B5), IGGPDEMV (A7), IGGPDEPT (B5), IGGPDIAX (B5), IGGPDIPT (B5), IGGPDUPG (B5), IGGPGET (BI), IGGPMCRA (A7), IGGPSMFS (BV), IGGPVMSC (A7)	IGGPDDCK, IGGPDEXA, IGGPDLXT, IGGPDOPN, IGGPERAS
IGGPDEXA	IGGPEXT (AZ)	IGGPDLXT
IGGPDLXT	IGGPARFS (AT)	—
IGGPDOPN	—	—
Internal		
IGGPDDCK	—	IGGPDLXT
IGGPERAS	IGGPRPLF (A3), IGGPRPLM (A3)	IGGPDLXT
IGG0CLBH		
External		
IGGPDAVO	IGGPMOD (AV)	—
IGGPDEFA	IGGPDGDS (A4), IGGPDUND (B8), IGGPSMFA (BV)	IGGPDAIN, IGGPDAVO

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPDAIN	IGGPAXCI (AG)	IGGPDANL
IGGPDANL	—	—
IGG0CLBI		
External		
IGGPGET	IGGPRAG (BO)	IGGPXIO, IGGPUCCT
IGGPTNXO	—	—
IGGPTXO	IGGPARFS (AT), IGGPEXT (AZ), IGGPMOD (AV)	—
IGGPUCRS	—	IGGPUCVT, IGGPGET
Internal		
IGGPUCCT	—	—
IGGPUCVT	—	—
IGGPXIO	IGGPIORA (AG)	—
IGG0CLBJ		
External		
IGGPGDSP	IGGPEXT (AZ)	IGGPGUDS
Internal		
IGGPGUDS	IGGPGET (BI)	—
IGG0CLBK		
External		
IGGPLDCS	—	IGGPLDCE
IGGPLSP	IGGPGET (BI), IGGPSMFL (BV)	IGGPLBVC, IGGPLDCS, IGGPLEMP, IGGPLSMP
Internal		
IGGPLBVC	—	—
IGGPLDAS	—	—
IGGPLDCE	IGGPEXT (AZ)	IGGPLDAS, IGGPLSMS
IGGPLEMP	—	IGGPLBVC
IGGPLSMP	—	—
IGGPLSMS	—	IGGPLDAS
IGG0CLBL		
External		
IGGPDELS	IGGPEXT (AZ)	IGGPDLCD, IGGPDLEX IGGPDLSC, IGGPDLSH, IGGPDLVC, IGGPDLVM
IGGPDLET	IGGPDEL (BG), IGGPDFM1 (AI), IGGPF4RD (BU), IGGPF4WR (BU), IGGPGET (BI), IGGPMOD (AV), IGGPRAOP (AD)	—
IGGPDLVM	—	IGGPVMV
IGGPDUCB	—	—

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPDLCD	IGGPEXT (AZ)	—
IGGPDLEX	—	—
IGGPDLMF	IGGPLSP (BK)	—
IGGPDLSC	—	—
IGGPDLSH	IGGPMOD (AV)	—
IGGPDLVC	IGGPEXT (AZ)	IGGPDLET, IGGPDLMF
IGGPVMV	—	—
IGG0CLBM		
External		
IGGPCKAU	IGGPINMD (B6), IGGPSPSC (B6)	IGGPBSGT, IGGPCKEX, IGGPLVST, IGGPPWGT, IGGPPWVR
Internal		
IGGPBSGT	IGGPGET (BI)	—
IGGPCKEX	IGGPEXT (AZ), IGGPRACS (DC)	—
IGGPCLGT	IGGPEXT (AZ), IGGPGET (BI) IGGPRACK (DC)	IGGPBSGT, IGGPCKEX
IGGPLVST	—	—
IGGPPWGT	IGGPGET (BI), IGGPRPLF (A3), IGGPRPLM (A3), IGGPWTSO (B6)	IGGPCKEX
IGGPPWVR	IGGPRACT (DC), IGGPRACV (DC)	IGGPBSGT, IGGPCKCC, IGGPCKEX, IGGPCLGT
IGG0CLBN		
External		
IGGPALMD	IGGPARFS (AT), IGGPMOD (AV)	IGGPALBT
IGGPALSV	IGGPARFS (AT)	—
IGGPALVR	IGGPALEC (BE), IGGPEXT (AZ), IGGPMOD (AV)	IGGPALPL, IGGPALVE, IGGPALVO
IGGPVRD	IGGPGET (BI)	IGGPVRCV
Internal		
IGGPACHR	—	—
IGGPALBT	—	—
IGGPALPL	—	—
IGGPALVE	IGGPGET (BI), IGGPMOD (AV)	IGGPALPL
IGGPALVO	—	—
IGGPVRCV	IGGPF4DQ (BU), IGGPF4RD (BU), IGGPF4WR (BU)	IGGPACHR

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLBO		
External		
IGGPRAG	—	IGGPRAOR, IGGPRAX, IGGPXRIO
IGGPRAPA	IGGPARFS (AT), IGGPPAD (AG), IGGPRAPV (DA)	IGGPRAOR, IGGPRAPC, IGGPRARA, IGGPRAX
IGGPRAPD	IGGPRAPV (DA)	IGGPRAOR, IGGPRARC, IGGPRAX, IGGPXRIO
IGGPRAPU	IGGPRAPV (DA)	IGGPRAOR, IGGPRAX, IGGPXRIO
IGGPRARC	—	IGGPXRIO
IGGPRASC	—	IGGPRAOR, IGGPRARA, IGGPRARC, IGGPRAX
IGGPXRIO	—	IGGPRAEA
Internal		
IGGPRACC	—	IGGPXRIO
IGGPRAEA	IGGPEMIO (AF), IGGPEMSG (AF)	—
IGGPRAOR	IGGPRAOP (AD)	—
IGGPRAPC	IGGPARFS (AT)	—
IGGPRARA	—	IGGPRACC, IGGPRARC, IGGPXRIO
IGGPRAX	—	IGGPRACC
IGG0CLBP		
External		
IGGPRET1	—	—
IGGPSPAC	IGGPF4DQ (BU), IGGPF4RD (BU), IGGPF4WR (BU), IGGPISCI (AG), IGGPPUPC (AG)	IGGPDALL, IGGPDEXT, IGGPINEX, IGGPOBTN, IGGPSRH1, IGGPSRH2, IGGPWRIT
Internal		
IGGPDALL	—	IGGPGNAM
IGGPDEXT	—	IGGPGNAM
IGGPGNAM	—	—
IGGPINEX	—	—
IGGPOBTN	—	—
IGGPSRH1	IGGPEXT (AZ)	—
IGGPSRH2	IGGPEXT (AZ)	—
IGGPWRIT	IGGPMOD (AV)	—
IGG0CLBQ		
External		
IGGPLSTC	IGGPCKAU (BM), IGGPEXT (AZ), IGGPGET (BI)	—

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLBR		
External		
IGGPBMR	—	IGGPGETM, IGGPPART, IGGPPUTM
Internal		
IGGPGETM	IGGPGET (BI)	IGGPPUTM
IGGPPART	—	—
IGGPPUTM	IGGPPUPC (AG)	
IGG0CLBS		
External		
IGGPXEXT	—	—
IGGPXVAL	IGGPBMR (BR), IGGPGREC (BA), IGGPGVAL (BA), IGGPSGOP (AV)	IGGPXVOL
Internal		
IGGPXEXS	IGGPGREC (BA), IGGPGVAL (BA)	—
IGGPXVOL		
IGG0CLBT		
External		
IGGPXDGO	IGGPADGO (AW), IGGPBMR (BR), IGGPGREC (BA), IGGPGVAL (BA)	IGGPCSMP
IGGPXEL2	IGGPBMR (BR), IGGPDEL2 (AV), IGGPGREC (BA), IGGPGVAL (BA)	—
IGGPXLT2	IGGPALT2 (AX)	—
IGGPXMOD	—	—
Internal		
IGGPCSMP	IGGPAOCI (AG), IGGPPUPC (AG)	—
IGG0CLBU		
External		
IGGPF4DQ	—	—
IGGPF4RD	—	IGGPF4DQ
IGGPF4WR	—	IGGPF4DQ
IGG0CLBV		
External		
IGGPSMFA	IGGPGET (BI)	IGGPSMFC, IGGPSMFD, IGGPSMFE, IGGPSMFF
IGGPSMFF	IGGPARFS (AT)	—
IGGPSMFL	IGGPGET (BI), IGGPLDCS (BK)	IGGPSMFC
IGGPSMFR	—	IGGPSMFC
IGGPSMFS	—	IGGPSMFC, IGGPSMFE
IGGPSMFZ		

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPSMFC	IGGPARFS (AT)	—
IGGPSMFD	IGGPARFS (AT)	IGGPSMFZ
IGGPSMFE	IGGPARFS (AT)	IGGPSMFM
IGGPSMFM	IGGPGET (BI)	IGGPSMFP
IGGPSMFP	—	—
IGG0CLBW		
External		
IGGPDEIN	IGGPGVAL (BA)	IGGPRISE, IGGPSINK, IGGPSNK2
Internal		
IGGPDOWN	IGGPAXCI (AG), IGGPEXPD (AX), IGGPGNEX (AW), IGGPGVAL (BA), IGGPPREC (AW), IGGPSHNK (AX)	—
IGGPRISE	IGGPAXCI (AG), IGGPGVAL (BA), IGGPPREC (AW), IGGPSHNK (AX)	IGGPSINK, IGGPUPUP
IGGPSINK	IGGPEXPD (AX), IGGPSHNK (AX)	—
IGGPSNK2	IGGPAXCI (AG), IGGPGREC (BA), IGGPPDE (AG), IGGPPREC (AW), IGGPSMF (B3)	IGGPDOWN
IGGPUPUP	—	—
IGG0CLBX		
External		
IGGPDCIM	—	—
IGGPDSPC	—	IGGPDCCC, IGGPDCCO, IGGPDDCE
Internal		
IGGPDCCC	—	IGGPDCID, IGGPDCIM, IGGPDCPT, IGGPDPBI, IGGPITER
IGGPDCCI	—	IGGPDPBI
IGGPDCID	—	IGGPDCIM, IGGPMAXA
IGGPDCIM	—	—
IGGPDCPT	—	—
IGGPDDCE	IGGPGET (BI)	—
IGGPDPBI	—	—
IGGPITER	—	—
IGGPMAXA	—	—
IGG0CLBY		
External		
IGGPDRSP	—	IGGPDATA, IGGPDCIS, IGGPDDRT, IGGPDDSA, IGGPDDTC, IGGPDISA

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPDATA	—	—
IGGPDCIS	—	—
IGGPDDRT	—	—
IGGPDDSA	—	IGGPDATA, IGGPDDRT, IGGPDDTC
IGGPDDTC	—	—
IGGPDISA	—	—
IGG0CLBZ		
External		
IGGPCONV	IGGPF4RD (BU), IGGPF4WR (BU), IGGPMOD (AV), IGGPPUPC (AG), IGGPRCCR (AG), IGGPRPLF (A3), IGGPSCAT (AH)	IGGPGALO, IGGPVALI
IGGPGALO	—	—
IGGPVALI	—	—
IGG0CLB0		
External		
IGGPDCCO	IGGPPSPC (BX)	IGGPDCCI, IGGPDPBI
Internal		
IGGPDCCI	—	IGGPDPBI
IGGPDPBI	—	—
IGG0CLB1		
External		
IGGPUADD	IGGPAXCI (AG), IGGPEXT (AZ), IGGPGET (BI), IGGPMOD (AV), IGGPPAD (AG), IGGPPDE (AG)	IGGPUCOM, IGGPUEND
IGGPUDEL	IGGPEXT (AZ), IGGPGET (BI), IGGPMOD (AV), IGGPPDE (AG)	IGGPUCOM, IGGPUEND
Internal		
IGGPUCOM	IGGPEXT (AZ), IGGPGET (BI)	—
IGGPUEND	IGGPGET (BI)	—
IGG0CLB2		
External		
IGGPALPG	—	—
IGGPAUPG	IGGPARFS (AT), IGGPEXT (AZ), IGGPUADD (B1), IGGPUDEL (B1)	IGGPAEXA
Internal		
IGGPAEXA	—	—

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLB3		
External		
IGGPSMF	—	—
IGGPSMFG	IGGPARFS (AT), IGGPSMFZ (BV)	—
IGG0CLB4		
External		
IGGPDCRA	IGGPMODI (DA), IGGPWCAT (DA), IGGPWCRA (DA)	IGGPCACB, IGGPCCIO, IGGPCHIU, IGGPCI15, IGGPCRPL, IGGPCXWA, IGGPDCXT, IGGPFMT4, IGGPOCRA, IGGPRDEF, IGGPSBAL, IGGPSTRG
Internal		
IGGPCACB	—	—
IGGPCCIO	—	—
IGGPCHIU	—	—
IGGPCI15	—	—
IGGPCRPL	—	—
IGGPCXWA	—	—
IGGPDCXT	IGGPARFS (AT)	—
IGGPFMT4	IGGPF4RD (BU), IGGPF4WR (BU)	—
IGGPOCRA	—	—
IGGPRDEF	IGGPDEFS (AQ)	—
IGGPSBAL	IGGPSALL (AR)	—
IGGPSTRG	IGGPARFS (AT)	IGGPRDEF, IGGPSBAL
IGG0CLB5		
External		
IGGPDCLS	IGGPDLXT (BG), IGGPDUND (B8), IGGPGET (BI), IGGPPDE (AG), IGGPROBD (DD), IGGPSMFS (BV), IGGPVMSC (A7)	—
IGGPDEAX	IGGPDLXT (BG), IGGPGET (BI)	IGGPDBMD, IGGPDEXB, IGGPDIPT, IGGPDUPG
IGGPDEPT	IGGPDLXT (BG), IGGPGET (BI), IGGPPDE (AG), IGGPRPTH (DD), IGGPSMFS (BV)	IGGPDBMD
IGGPDIAX	IGGPDEXA (BG), IGGPDLXT (BG), IGGPDOPN (BG), IGGPGET (BI)	IGGPDBMD, IGGPDCLS, IGGPDEXB, IGGPDIPT, IGGPDUPG
IGGPDIPT	IGGPDLXT (BG), IGGPGET (BI), IGGPPDE (AG), IGGPSMFS (BV)	IGGPDBMD, IGGPDEXB
IGGPDUPG	IGGPDLXT (BG), IGGPUADD (B1), IGGPUDEL (B1)	—

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPDBMD	IGGPDLXT (BG), IGGPMOD (AV)	—
IGGPDEXB	IGGPDLXT (BG), IGGPEXT (AZ)	—
IGG0CLB6		
External		
IGGPCKCC	IGGPGET (BI)	—
IGGPINMD	IGGPGET (BI)	—
IGGPSPSC	IGGPCKEX (BM), IGGPGET (BI)	—
IGGPWTSO	—	IGGPGTSO
Internal		
IGGPGTSO	IGGPRPLF (A3), IGGPRPLM (A3)	—
IGG0CLB7		
External		
IGGPRUS	IGGPARFS (AT), IGGPEXT (AZ), IGGPGET (BI), IGGPLOC (AZ), IGGPMOD (AV), IGGPSSCR (BF), IGGPTNXO (BI), IGGPTXO (BI)	IGGPFRWK
Internal		
IGGPFRWK	IGGPARFS (AT), IGGPGET (BI), IGGPLOC (AZ)	—
IGG0CLB8		
External		
IGGPCNBO	IGGPGET (BI), IGGPMOD (AV)	IGGPDFBI
IGGPDFBO	IGGPARFS (AT), IGGPDUND (B8), IGGPGET (BI), IGGPVMSC (A7)	IGGPBCAN, IGGPBDSH, IGGPBDSS, IGGPBNUN, IGGPBSCR, IGGPDFBI
IGGPDUND	IGGPGET (BI), IGGPPDE (AG)	IGGPDEUN
Internal		
IGGPBCAN	IGGPGET (BI), IGGPMOD (AV)	—
IGGPBDSH	IGGPMOD (AV)	—
IGGPBDSS	IGGPMOD (AV)	—
IGGPBNUN	IGGPSSCR (BF)	—
IGGPBSCR	—	—
IGGPDFBI	—	—
IGG0CLB9		
External		
IGGPPAIX	IGGPDRDA (AN), IGGPDUND (B8), IGGPPSEM (AN)	IGGPDCBG, IGGPIGDC, IGGPMODC, IGGPPRPW, IGGPRGBC
IGGPPATH	—	IGGPBAMC, IGGPBAWP, IGGPCGGC, IGGPIGDC, IGGPRPW, IGGPSCRG, IGGPWSMF

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
Internal		
IGGPBAMC	IGGPMOD (AV)	—
IGGPBAWP	IGGPAOCI (AG), IGGPMOD (AV), IGGPPDE (AG)	IGGPINAE, IGGPINBE
IGGPCGGC	IGGPEXT (AZ)	IGGPEXGC
IGGPDCBG	IGGPDCCE (AN)	—
IGGPEXGC	IGGPEXT (AZ), IGGPGET (BI)	—
IGGPIGDC	—	—
IGGPINAE	IGGPEXT (AZ)	—
IGGPINBE	—	—
IGGPMODC	IGGPAOCI (AG), IGGPGET (BI), IGGPMOD (AV)	—
IGGPPRPW	IGGPCKAU (BM), IGGPGET (BI)	—
IGGPRGBC	IGGPMOD (AV)	—
IGGPRPW	—	—
IGGPSCRG	—	—
IGGPWSMF	IGGPSMFA (BV)	—
IGG0CLCA		
External		
IGG0CLCA	—	—
IGG0CLC9		
External		
IGG0CLC9	IGGPACDV (AB), IGGPRET1 (BP)	BLDCCA, IGGPRCU
Internal		
BLDCCA	—	—
IGGPRCU	IGGPARFS (AT), IGGPEMSG (AF), IGGPRARC (BO), IGGPRPLF (A3)	—
IGG0CLDA		
External		
IGGPMODI	IGGPMOD (AV)	—
IGGPRAPV	IGGPF4DQ (BU), IGGPF4RD (BU), IGGPF4WR (BU), IGGPXIO (AG)	—
IGGPWCAT	IGGPARFS (AT), IGGPPAD (AG)	—
IGGPWCRA	IGGPXRIO (BO)	—

Procedure Calls Directory (continued)

Calling Module and its Procedures	External procedures called (Procedures outside the module) (Abbreviated module name in parentheses is IGG0CLxx)	Internal procedures called (Procedures within the module)
IGG0CLDD		
External		
IGGPALRI	—	—
IGGPALRN	—	—
IGGPRDEF	—	—
IGGPRDEL	—	—
IGGPRDLC	—	IGGPRDEL
IGGPRFDC	IGGPGALO (BZ)	IGGPRDEL
IGGPRPTH	—	IGGPRDEL
IGGPROBD	—	IGGPRDEL
IGG0CLDC		
External		
IGGPRACG	—	IGGPRACK
IGGPRACT	—	IGGPRACK
IGGPRACV	IGGPBSGT (BM)	IGGPRACK, IGGPRACX
Internal		
IGGPRACE	IGGPEXT (AZ)	IGGPRACS
IGGPRACK	—	IGGPRACE
IGGPRACS	—	—
IGGPRACX	IGGPGRE (A3)	—

Procedure Called-By Directory

This table lists procedures that call an OS/VS2 Catalog Management procedure. See "Procedure Calls Directory" to determine the calling procedures within the module.

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLAA	
Procedure:	
IGGPSLEN	IGGPSLOC (AM)
IGG0CLAB	
Procedure:	
IGGPACDV	IGG0CLC9 (C9)
IGG0CLAC	
Procedure:	
IGGPMCO	IGGPACDV (AB)
IGG0CLAD	
Procedures:	
IGGPMCO2	IGGPDCOC (AE), IGGPMCO (AC)
IGGPRAOP	IGGPDLET (BL), IGGPRAOR (BO)
IGG0CLAE	
Procedure:	
IGGPDCME	IGGPDEFC (AS)
IGGPMEBM	IGGPDCSP (AS)
IGG0CLAF	
Procedures:	
IGGPDELC	IGGPCDVR (AT)
IGGPEMIO	IGGPIORA (AG), IGGPRAEA (BO)
IGGPEMSG	IGGPIORA (AG), IGGPRAEA (BO), IGGPRCU (C9)
IGG0CLAG	
Procedures:	
IGGPAOCI	IGGPBAWP (B9), IGGPCSMP (BT), IGGPDCCE (AN), IGGPMODC (B9)
IGGPAXCI	IGGPADGO (AW), IGGPAGOP (AW), IGGPDAIN (BH), IGGPDCPC (AP), IGGPDEFB (A4), IGGPDEFS (AQ), IGGPDEFX (AO), IGGPDOWN (BW), IGGPMBGO (AX), IGGPMVAR (AX), IGGPRISE (BW), IGGPSNK2 (BW)
IGGPCCCR	IGGPCMRR (A9), IGGPCONV (BZ), IGGPDDBO (A9), IGGPDEFC (AS), IGGPDELC (AF)
IGGPIORA	IGGPXIO (BI)
IGGPISCI	IGGPDEFS (AQ), IGGPSALL (AR), IGGPSPAC (BP)
IGGPPAD	IGGPALNM (BD), IGGPDCCE (AN), IGGPDCPC (AP), IGGPDEFB (A4), IGGPDEFS (AQ), IGGPPREC (AW), IGGPRAPA (BO), IGGPUADD (B1), IGGPWCAT (DA)
IGGPPDE	IGGPALNM (BD), IGGPBAWP (B9), IGGPDCLS (B5), IGGPDEFX (AO), IGGPDELB (A5), IGGPDELC (AF), IGGPDEL2 (AV), IGGPDEPT (B5), IGGPDIPT (B5), IGGPDONE (A5), IGGPDUND (B8), IGGPSNK2 (BW), IGGPUADD (B1), IGGPUDEL (B1)

Procedure Called-By Directory (continued)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGGPPUPC	IGGPALT (BD), IGGPCONV (BZ), IGGPCSMP (BT), IGGPDF4T (A7), IGGPDGDS (A4), IGGPFDSP (AI), IGGPPREC (AW), IGGPPUTM (BR), IGGPSPAC (BP)
IGGPRCCR	IGGPCONV (BZ)
IGGPXIO	IGGPDCPR (AE), IGGPDELC (AF), IGGPRAPV (DA)
IGG0CLAH	
Procedures:	
IGGPCSDO	IGGPCSCP (A3)
IGGPQNQC	IGGPALNM (BD), IGGPDEDE (AL)
IGGPSCAT	IGGPACDV (AB), IGGPCDVR (AT), IGGPCONV (BZ), IGGPDEFS (AQ), IGGPDELC (AF)
IGGPCS2	IGGPGLOC (A2)
IGG0CLAI	
Procedures:	
IGGPDFM1	IGGPDELT (BL)
IGGPFDSP	IGGPDELC (AF)
IGG0CLAJ	
Procedure:	
IGGPDBDI	IGGPDSCB (AN)
IGG0CLAK	
Procedure:	
IGGPDCMB	IGGPDBDI (AJ)
IGG0CLAL	
Procedures:	
IGGPDEF	IGGPCDVR (AT)
IGGPDTIM	IGGPDBDI (AJ), IGGPDEFS (AQ)
IGG0CLAM	
Procedures:	
IGGPDBVC	IGGPDELC (AF), IGGPGLOC (A2)
IGGPSLEL	IGGPGTAB (A2), IGGPSLEN (AA), IGGPSLR (AA), IGGPSLY (AA)
IGGPSLOC	IGGPACDV (AB)
IGG0CLAN	
Procedures:	
IGGPDCCE	IGGPDCBG (B9), IGGPDCPC (AP)
IGGPDRDA	IGGPPAIX (B9)
IGGPDSCB	IGGPDEF (AL)
IGGPPSEM	IGGPPAIX (B9)
IGG0CLAP	
Procedure:	
IGGPDCDA	IGGPDSCB (AN)

Procedure Called-By Directory (continued)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLAQ	
Procedure:	
IGGPDEFS	IGGPCDVR (AT), IGGPDCSP (AS), IGGPDSPO (AJ), IGGPRDEF (B4)
IGG0CLAR	
Procedure:	
IGGPSALL	IGGPALSA (BE), IGGPCDSD (A6), IGGPCSAL (BB), IGGPDCNV (AJ), IGGPDCSP (AS), IGGPDSPO (AJ), IGGPSBAL (B4)
IGG0CLAS	
Procedures:	
IGGPDCRC	IGGPDCME (AE)
IGGPDEFC	IGGPDSCB (AN)
IGG0CLAT	
Procedures:	
IGGPARFS	IGGPALF1 (BD), IGGPALGV (BD), IGGPALMD (BN), IGGPALSV (BN), IGGPALT (BD), IGGPALVL (BE), IGGPAUPG (B2), IGGPBJFB (A6), IGGPCSAL (BB), IGGPDBCV (AK), IGGPDBVO (AK), IGGPDCDS (AF), IGGPDCNV (AJ), IGGPDCOC (AE), IGGPDCSP (AS), IGGPDCXT (B4), IGGPDEFS (AQ), IGGPDELB (A5), IGGPDEVG (A7), IGGPDEXD (AJ), IGGPDFBO (B8), IGGPDFRE (A8), IGGPDFRS (A8), IGGPDGBC (A5), IGGPDGDS (A4), IGGPDLXT (BG), IGGPDMOP (AK), IGGPDOG (A3), IGGPDSEX (AJ), IGGPDSXT (AQ), IGGPFDSP (AI), IGGPFRWK (B7), IGGPTXO (BI), IGGPRAPA (BO), IGGPRAPC (BO), IGGPRCU (C9), IGGPRUS (B7), IGGPSMFC (BV), IGGPSMFD (BV), IGGPSMFE (BV), IGGPSMFF (BV), IGGPSMFG (B3), IGGPSTRG (B4), IGGPVMSC (A7), IGGPWCAT (DA)
IGGPCDVR	IGGPACDV (AB), IGGPDDBO (A9)
IGG0CLAU	
Procedure:	
IGGPSALS	IGGPSALL (AR)

Procedure Called-By Directory (continued)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLAV	
Procedures:	
IGGPDEL2	IGGPXEL2 (BT)
IGGPMOD	IGGPALMD (BN), IGGPALSA (BE), IGGPALVE (BN), IGGPALVR (BN), IGGPANVV (BE), IGGPBAMC (B9), IGGPBAWP (B9), IGGPBCAN (B8), IGGPBCFI (AO), IGGPBDSH (B8), IGGPBDSS (B8), IGGPCNBO (B8), IGGPCONV (BZ), IGGPDAVO (BH), IGGPDBMD (B5), IGGPDCME (AE), IGGPDEDD (A7), IGGPDEFX (AO), IGGPDELB (A5), IGGPDESH (A7), IGGPDEVG (A7), IGGPDGDG (A4), IGGPDLET (BL), IGGPDLSH (BL), IGGPDMOP (A6), IGGPDMSG (A7), IGGPDONE (A5), IGGPDSMD (A6), IGGPERUG (A4), IGGPFDSP (AI), IGGPMEXT (BB), IGGPMODC (B9), IGGPMODI (DA), IGGPMVOL (BB), IGGPTXO (BI), IGGPRGBC (B9), IGGPRUS (B7), IGGPSALL (AR), IGGPUADD (B1), IGGPUDEL (B1), IGGPWRIT (BP)
IGGPPUPD	IGGPACDV (AB)
IGGPSGOP	IGGPMGO (AX), IGGPXVAL (BS)
IGG0CLAW	
Procedures:	
IGGPADGO	IGGPXDGO (BT)
IGGPGNEX	IGGPDOWN (BW), IGGPMBGO (AX), IGGPMVAR (AX)
IGGPIGOP	IGGPMGO (AX)
IGGPPREC	IGGPDEL2 (AV), IGGPDOWN (BW), IGGPGREC (BA), IGGPRISE (BW), IGGPSFPL (AV), IGGPSNK2 (BW)
IGG0CLAX	
Procedures:	
IGGPALT2	IGGPXLT2 (BT)
IGGPDGO	IGGPDEL2 (AV)
IGGPDGOP	IGGPDEL2 (AV)
IGGPEXPD	IGGPDOWN (BW), IGGPSINK (BW)
IGGPMGO	IGGPAGOP (AW)
IGGPSHNK	IGGPDOWN (BW), IGGPRISE (BW), IGGPSINK (BW)
IGG0CLAY	
Procedure:	
IGGPSCNC	IGGPACDV (AB), IGGPEXT (AZ), IGGPMOD (AV)

Procedure Called-By Directory (continued)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLAZ	
Procedures:	
IGGPEXT	IGGPALGV (BD), IGGPALIX (BE), IGGPALVA (BE), IGGPALVL (BE), IGGPALVR (BN), IGGPAUPG (B2), IGGPCGGC (B9), IGGPCKEX (BM), IGGPCLGT (BM), IGGPDALL (A5), IGGPDCDS (AF), IGGPDEFX (AQ), IGGPDELB (A5), IGGPDELC (AF), IGGPDELS (BL), IGGPDEMV (A7), IGGPDEVG (A7), IGGPDEXA (BG), IGGPDEXB (B5), IGGPDGDS (A4), IGGPDLCD (BL), IGGPDLVC (BL), IGGPDMSG (A7), IGGPDONE (A5), IGGPERUG (A4), IGGPEXGC (B9), IGGPFDSP (AI), IGGPGDSP (BJ), IGGPGGE (A2), IGGPGTAB (A2), IGGPINAE (B9), IGGPINIT (BC), IGGPLDCE (BK), IGGPLSTC (BQ), IGGPTXO (BI), IGGPRACE (DC), IGGPRUS (B7), IGGPSALL (AR), IGGPSALS (AU), IGGPSLCG (AA), IGGPSLEI (AM), IGGPSLEL (AM), IGGPSLER (AM), IGGPSLR (AA), IGGPSLY (AA), IGGPSRH2 (BP), IGGPSRH2 (BP), IGGPSVOL (BC), IGGPUADD (B1), IGGPUCOM (B1), IGGPUDEL (B1), IGGPUPDE (BB)
IGGPLOC	IGGPACDV (AB), IGGPFRWK (B7), IGGPRUS (B7)
IGG0CLA2	
Procedures:	
IGGPGDGL	IGGPSLOC (AM)
IGGPGLOC	IGGPSLOC (AM)
IGG0CLA3	
Procedures:	
IGGPRE	IGGPRACX (DC)
IGGPRPLF	IGGPCMRR (A9), IGGPCONV (BZ), IGGPDDBO (A9), IGGPDELC (AF), IGGPDYNA (A3), IGGPERAS (BG), IGGPGTSO (B6), IGGPPWGT (BM), IGGPRCU (C9)
IGGPRPLM	IGGPDYNA (A3), IGGPERAS (BG), IGGPGTSO (B6), IGGPPWGT (BM)
IGG0CLA4	
Procedures:	
IGGPDEFB	IGGPCDVR (AT)
IGGPDEFX	IGGPCDVR (AT)
IGGPDGDS	IGGPDEFA (BH)
IGG0CLA5	
Procedures:	
IGGPDALL	IGGPDELC (AF)
IGGPDELB	IGGPCDVR (AT)
IGGPDELX	IGGPCDVR (AT)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLA6	
Procedures:	
IGGPBJFB	IGGPDEFS (AQ)
IGGPCBPT	IGGPDEFS (AQ)
IGGPCRTC	IGGPDEFS (AQ)
IGGPDALL	IGGPDEL (BG)
IGGPDFS2	IGGPDEFS (AQ)
IGG0CLA7	
Procedures:	
IGGPDEMV	IGGPDEL (BG)
IGGPDF4T	IGGPDLER (AF)
IGGPMCRA	IGGPDEL (BG)
IGGPVMSC	IGGPDCLS (B5), IGGPDEL (BG), IGGPDFBO (B8)
IGG0CLA8	
Procedure:	
IGGPDFRS	IGGPDBDI (AJ)
IGG0CLA9	
Procedure:	
IGGPCCBC	IGGPRPLM (A3)
IGG0CLBA	
Procedures:	
IGGPGREC	IGGPADGO (AW), IGGPAGOP (AW), IGGPDEL2 (AV), IGGPGREL (AW), IGGPGREP (AZ), IGGPMBGO (AX), IGGPMVAR (AX), IGGPSCNF (AZ), IGGPSFPL (AV), IGGPSGOP (AV), IGGPSNK2 (BW), IGGPXDGO (BT), IGGPXEL2 (BT), IGGPXEXS (BS), IGGPXVAL (BS)
IGGPGVAL	IGGPALT2 (AX), IGGPDEIN (BW), IGGPDOWN (BW), IGGPLOC2 (AZ), IGGPMVAR (AX), IGGPRISE (BW), IGGPUPGD (AZ), IGGPXDGO (BT), IGGPXEL2 (BT), IGGPXEXS (BS), IGGPXVAL (BS)
IGGPTSTS	IGGPSCNF (AZ), IGGPSFPL (AV)
IGG0CLBB	
Procedure:	
IGGPUPDE	IGGPUPD (AV)
IGG0CLBC	
Procedures:	
IGGPINIT	IGGPSSWD (BB), IGGPUPDE (BB)
IGGPSVOL	IGGPSSWD (BB), IGGPUPDE (BB)
IGG0CLBD	
Procedure:	
IGGPALT	IGGPCDVR (AT)

Procedure Called-By Directory (continued)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLBE	
Procedures:	
IGGPALEC	IGGPALVR (BN)
IGGPALVL	IGGPALT (BD)
IGGPANVV	IGGPALT (BD)
IGG0CLBF	
Procedures:	
IGGPSSCR	IGGPBNUN (B8), IGGPRUS (B7), IGGPVMSC (A7)
IGG0CLBG	
Procedures:	
IGGPDEL	IGGPCDVR (AT), IGGPDELB (A5), IGGPDGDS (A4), IGGPDLET (BL)
IGGPDEXA	IGGPDIAX (B5)
IGGPDLXT	IGGDBMD (B5), IGGPDCLS (B5), IGGPDEAX (B5), IGGPDEPT (B5), IGGPDEVG (A7), IGGPDEXB (B5), IGGPDF4T (A7), IGGPDIAX (B5), IGGPDIPT (B5), IGGPDUPG (B5), IGGPDUSC (A7)
IGGPDOPN	IGGPDCDS (AF), IGGPDIAX (B5), IGGPFDSP (AI)
IGG0CLBH	
Procedures:	
IGGPDAVO	IGGPANVV (BE)
IGGPDEFA	IGGPCDVR (AT)
IGG0CLBI	
Procedures:	
IGGPGET	IGGPALEC (BE), IGGPALIX (BE), IGGPALNM (BD), IGGPALT (BD), IGGPALVE (BN), IGGPBCAN (B8), IGGPBSGT (BM), IGGPCDVR (AT), IGGPCKCC (B6), IGGPCLGT (BM), IGGPCNBO (B8), IGGPDALL (A5), IGGPDBDI (AJ), IGGPDCLS (B5), IGGPDDCE (BX), IGGPDEAX (B5), IGGPDEDE (AL), IGGPDEFS (AQ), IGGPDEFX (AO), IGGPDEL (BG), IGGPDELB (A5), IGGPDELC (AF), IGGPDELX (A5), IGGPDEPT (B5), IGGPDFBO (B8), IGGPDGDS (A4), IGGPDIAX (B5), IGGPDIPT (B5), IGGPDLET (BL), IGGPDOG (A3), IGGPDONE (A5), IGGPDSPO (AJ), IGGPDUND (B8), IGGPEXGC (B9), IGGPFDSP (AI), IGGPFRWK (B7), IGGPGETM (BR), IGGPGGE (A2), IGGPGLOC (A2), IGGPGRE (A3), IGGPGREC (BA), IGGPGUDS (BJ), IGGPINMD (B6), IGGPISCI (AG), IGGPLSP (BK), IGGPLSTC (BQ), IGGPMODC (B9), IGGPPRPW (B9), IGGPPWGT (BM), IGGPQNQC (AH), IGGPRUS (B7), IGGPSALL (AR), IGGPSLCG (AA), IGGPSLEI (AM), IGGPSLR (AA), IGGPSLY (AA), IGGPSMFA (BV), IGGPSMFL (BV), IGGPSMFM (BV), IGGPSPSC (B6), IGGPSSWD (BB), IGGPUADD (B1), IGGPUALL (BB), IGGPUCOM (B1), IGGPUDEL (B1), IGGPUEND (B1), IGGPUPDE (BB), IGGPUPGD (AZ), IGGPVMSC (A7), IGGPVRD (BN)

Procedure Called-By Directory (continued)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGGPTNXO	IGGPDSPO (AJ), IGGPRUS (B7), IGGPUPDE (BB)
IGGPTXO	IGGPDSPO (AJ), IGGPRUS (B7), IGGPUPDE (BB)
IGGPUCRS	IGGPSCAT (AH)
IGG0CLBJ	
Procedure:	
IGGPGDSP	IGGPACDV (AB)
IGG0CLBK	
Procedures:	
IGGPLDCS	IGGPSMFL (BV)
IGGPLSP	IGGPACDV (AB), IGGPDEFS (AQ), IGGPDLMF (BL)
IGG0CLBL	
Procedures:	
IGGPDELS	IGGPCDVR (AT)
IGGPDLET	IGGPFDSP (AI)
IGGPDLVM	IGGPFDSP (AI)
IGGPDUCB	IGGPFDSP (AI)
IGG0CLBM	
Procedure:	
IGGPBSGT	IGGPRACV (DC)
IGGPCKAU	IGGPACDV (AB), IGGPCDVR (AT), IGGPDEFS (AQ), IGGPDOG (A3), IGGPLSTC (BQ), IGGPPRPW (B9)
IGGPCKEX	IGGPSPSC (B6)
IGG0CLBN	
Procedures:	
IGGPALMD	IGGPALT (BD)
IGGPALVR	IGGPALVL (BE)
IGGPVRD	IGGPALVL (BE)
IGG0CLBO	
Procedures:	
IGGPRAG	IGGPGET (BI)
IGGPRAPA	IGGPPAD (AG)
IGGPRAPD	IGGPPDE (AG)
IGGPRAPU	IGGPPAD (AG), IGGPPUPC (AG)
IGGPRARC	IGGPRCU (C9)
IGGPRASC	IGGPISCI (AG)
IGGPXRIO	IGGPWCRA (DA)
IGG0CLBP	
Procedures:	
IGGPRET1	IGG0CLC9 (C9)
IGGPSPAC	IGGPCSAL (BB)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLBQ	
Procedure:	
IGGPLSTC	IGGPCDVR (AT)
IGG0CLBR	
Procedure:	
IGGPBMR	IGGPMEBM (AE), IGGPEDS (AU), IGGPSALS (AU), IGGPSSCR (BF), IGGPXDGO (BT), IGGPXEL2 (BT), IGGPXVAL (BS)
IGG0CLBS	
Procedures:	
IGGPXEXT	IGGPEXT (AZ)
IGGPXVAL	IGGPGVAL (BA)
IGG0CLBT	
Procedures:	
IGGPXDGO	IGGPSFPL (AV)
IGGPXEL2	IGGPSFPL (AV)
IGGPXLT2	IGGPSFPL (AV)
IGGPXMOD	IGGPMOD (AV)
IGG0CLBU	
Procedures:	
IGGPF4DQ	IGGPDFM1 (AI), IGGPDF4T (A7), IGGPF4PR (AQ), IGGPRAPV (DA), IGGPRCTV (A7), IGGPSPAC (BP), IGGPVRCV (BN)
IGGPF4RD	IGGPCONV (BZ), IGGPDCBO (AE), IGGPDELC (AF), IGGPDFM1 (AI), IGGPDF4T (A7), IGGPDLET (BL), IGGPFMT4 (B4), IGGPF4PR (AQ), IGGPRAPV (DA), IGGPRCTV (A7), IGGPSPAC (BP), IGGPVRCV (BN)
IGGPF4WR	IGGPCONV (BZ), IGGPDCBO (AE), IGGPDELC (AF), IGGPDF4T (A7), IGGPDLET (BL), IGGPFMT4 (B4), IGGPF4PR (AQ), IGGPRAPV (DA), IGGPRCTV (A7), IGGPSPAC (BP), IGGPVRCV (BN)
IGG0CLBV	
Procedures:	
IGGPSMFA	IGGPALT (BD), IGGPDBDI (AJ), IGGPDCOC (AE), IGGPDEFA (BH), IGGPDEFB (A4), IGGPDEFX (AO), IGGPDOMF (AK), IGGPUPDE (BB), IGGPWSMF (B9)
IGGPSMFL	IGGPLSP (BK), IGGPUPDE (BB)
IGGPSMFR	IGGPALNM (BD)
IGGPSMFS	IGGPDCLS (B5), IGGPDEL (BG), IGGPDELB (A5), IGGPDEPT (B5), IGGDIPT (B5), IGGPDONE (A5)
IGGPSMFZ	IGGPSMFG (B3)
IGG0CLBW	
Procedure:	
IGGPDEIN	IGGPMVAR (AX)

Procedure Called-By Directory (continued)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLBX	
Procedures:	
IGGPDSPC	IGGPDRDA (AN)
IGGPPSPC	IGGPDCCO (B0)
IGG0CLBY	
Procedure:	
IGGPDRSP	IGGPDRDA (AN)
IGG0CLBZ	
Procedure:	
IGGPCONV	IGGPCDVR (AT)
IGGPGALO	IGGPRFDC (DD)
IGG0CLB0	
Procedure:	
IGGPDCCO	IGGPDSPC (BX)
IGGPFPDA	IGGPDDEP (AL)
IGGPCMKY	IGGPDRPG (AL)
IGGPDCNV	IGGPDPDI (AJ)
IGGPDPBI	IGGPDCCC (BX)
IGGPITER	IGGPDCCC (BX)
IGG0CLB1	
Procedures:	
IGGPUADD	IGGPAUPG (B2), IGGPDBDI (AJ), IGGPDUPG (B5)
IGGPUDEL	IGGPAUPG (B2), IGGPDUPG (B5)
IGG0CLB2	
Procedure:	
IGGPAUPG	IGGPALT (BD)
IGG0CLB3	
Procedures:	
IGGPSMF	IGGPALT (BD), IGGPDEL2 (AV), IGGPPREC (AW), IGGPSNK2 (BW)
IGGPSMFG	IGGPALT (BD), IGGPGREC (BA), IGGPSCNF (AZ), IGGPSFPL (AV)
IGG0CLB4	
Procedure:	
IGGPDCRA	IGGPDCSP (AS), IGGPDEFS (AQ)

Procedure Called-By Directory (continued)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLB5	
Procedures:	
IGGPDCLS	IGGPDEL (BG)
IGGPDEAX	IGGPDEL (BG)
IGGPDEPT	IGGPDEL (BG)
IGGPDIAX	IGGPDEL (BG)
IGGPDIPT	IGGPDEL (BG)
IGGPDUPG	IGGPDEL (BG)
IGG0CLB6	
Procedures:	
IGGPINMD	IGGPCKAU (BM)
IGGPSPSC	IGGPCKAU (BM)
IGGPWTSO	IGGPPWGT (BM)
IGG0CLB7	
Procedure:	
IGGPRUS	IGGPUPD (AV)
IGG0CLB8	
Procedures:	
IGGPCNBO	IGGPDSPO (AJ)
IGGPDFBO	IGGPDBDI (AJ)
IGGPDUND	IGGPDCBO (AE), IGGPDCLS (B5), IGGPDEFA (BH), IGGPDFBO (B8), IGGPPAIX (B9)
IGG0CLC9	
Procedure:	
IGG0CLC9	IGG0CLA1 (A1)
IGG0CLDA	
Procedures:	
IGGPMODI	IGGPDCRA (B4)
IGGPRAPV	IGGPRAPA (BO), IGGPRAPD (BO), IGGPRAPU (BO)
IGGPWCAT	IGGPDCRA (B4)
IGGPWCRA	IGGPDCRA (B4)
IGG0CLDC	
Procedures:	
IGGPRACG	IGGPDGDS (A4)
IGGPRACK	IGGPCLGT (BM)
IGGPRACS	IGGPCKEX (BM)
IGGPRACT	IGGPPWVR (BM)
IGGPRACV	IGGPPWVR (BM)

Procedure Called-By Directory (continued)

Called module and its external procedures	Calling procedure (in module): (Abbreviated module name in parentheses is IGG0CLxx)
IGG0CLDD	
Procedures:	
IGGPALRI	IGGPCDVR (AT)
IGGPALRN	IGGPALNM (BD)
IGGPRDEF	IGGPDEDE (AL), IGGPCDVR (AT)
IGGPRDEL	IGGPCDVR (AT)
IGGPRDLC	IGGPCDVR (AT)
IGGPRFDC	IGGPCDVR (AT)
IGGPROBD	IGGPDCLS (B5)
IGGPRPTH	IGGPDIPT (B5)

DATA AREAS

"Data Areas" describes the OS/VS2 catalog and its records. "Data Areas" also describes each OS/VS2 catalog management control block, and shows the relationships between the catalog managment control blocks.

Catalog Record Descriptions

Each 512-byte catalog record in the low-key range of the catalog occupies a full control interval and contains the number of the control interval in which it resides. Each catalog record also contains the record type of the record. The low-address part of the catalog is made up of the following types of records:

- Alias name record, which relates an alias name to the nonVSAM data set's or user catalog's True Name record. This record is record type "X."
- Control record, which describes the free control intervals in the low-address part of the catalog. The Control record is the fourth record in the catalog. This record is record type "L."
- Free record, which marks the control interval in which it resides as available for use as another kind of catalog record. There is one Free record for each previously assigned control interval that is available for use. This record is record type "F."
- Cluster record, which describes a VSAM data-set cluster or a pagespace (conceptually an entry-sequenced VSAM data set). This record contains the control interval number of a Data record and, if the VSAM data set is a key-sequenced data set, the control interval number of an Index record. There is one Cluster record for each VSAM cluster cataloged. This record is record type "C."
- Data and Index records, which describe data sets and indexes. A Data record can also describe a pagespace (conceptually an entry-sequenced VSAM data set). There is one Data or Index record for each data set or index cataloged. These records are record types "D" and "I."
- Alternate index record, which relates the alternate index to its associated base cluster and also to any paths over it. This record is record type "G."
- Path record, which relates a base cluster and possibly an alternate index. This record is record type "R."
- Upgrade set record, which relates the data and index components of the alternate indexes that comprise the upgrade set. This record is record type "Y."
- Generation data group (GDG) base record, which contains the control interval number and generation level (absolute name) of each generation data set in the group. This record is record type "B."
- NonVSAM record, which describes a data set organized differently from VSAM. There is one NonVSAM record for each nonVSAM data set cataloged. This record is record type "A."
- User-Catalog record, which describes a user catalog. There is one User-Catalog record for each user catalog connected to the master catalog. This record is record type "U."

- Volume record, which describes each VSAM data space on a volume, the data sets that reside in the data space, and the space available within the data space. There is one Volume record for each volume controlled by this catalog. This record is record type "V."
- Extension record, which contains overflow information from another catalog record. There are as many Extension records as are required to contain overflow information. This record is record type "W" when it is an extension of a Volume record; it is record type "E" when it is an extension of any other catalog record.

The Cluster, Data, Index, Alternate Index, Path, Upgrade Set, NonVSAM, Extension, Alias, GDG Base, and User-Catalog records have a common general format. Figure 39 shows the general format for these records.

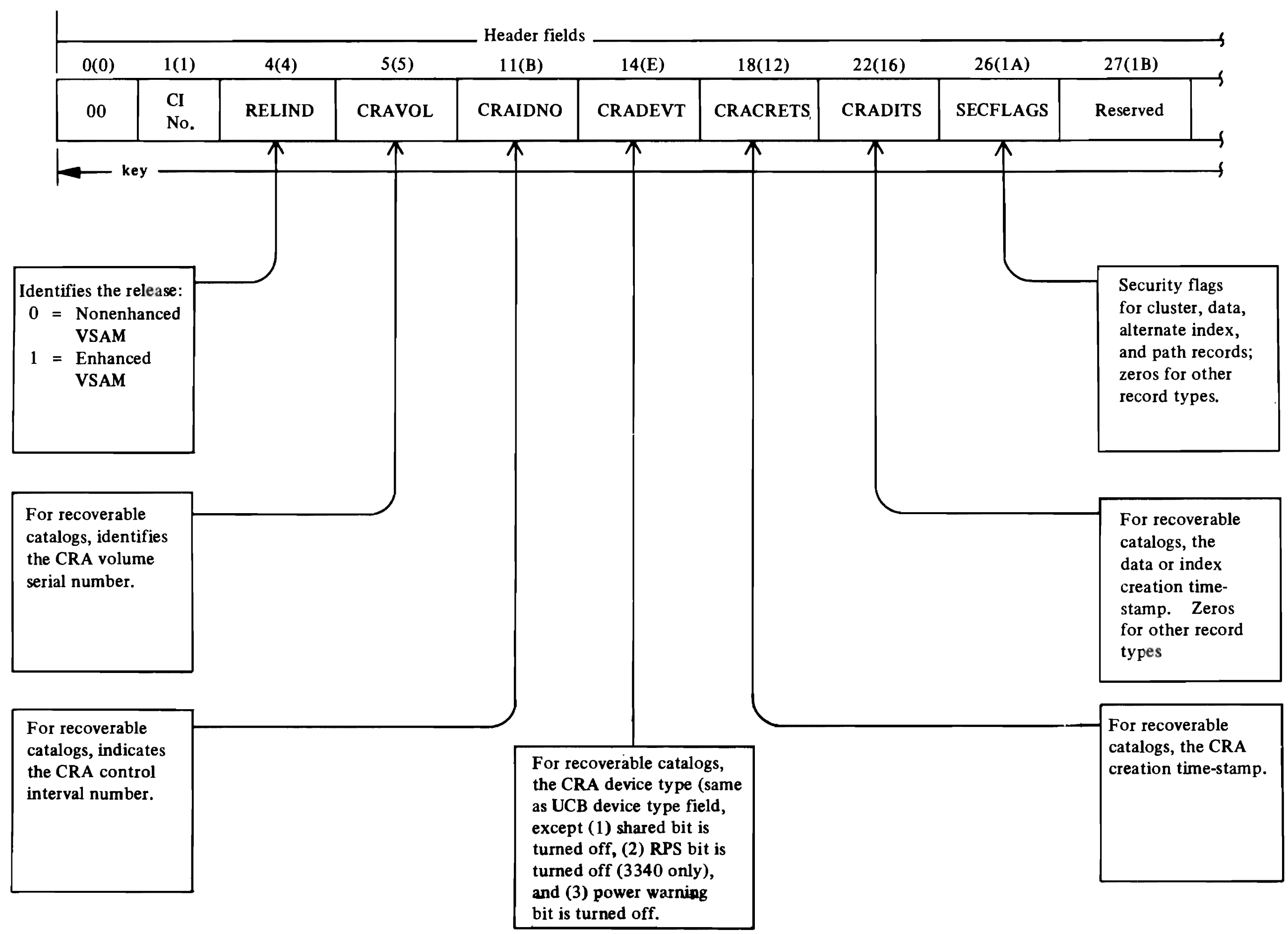


Figure 39 (Part 1 of 2). Catalog Record—General Format

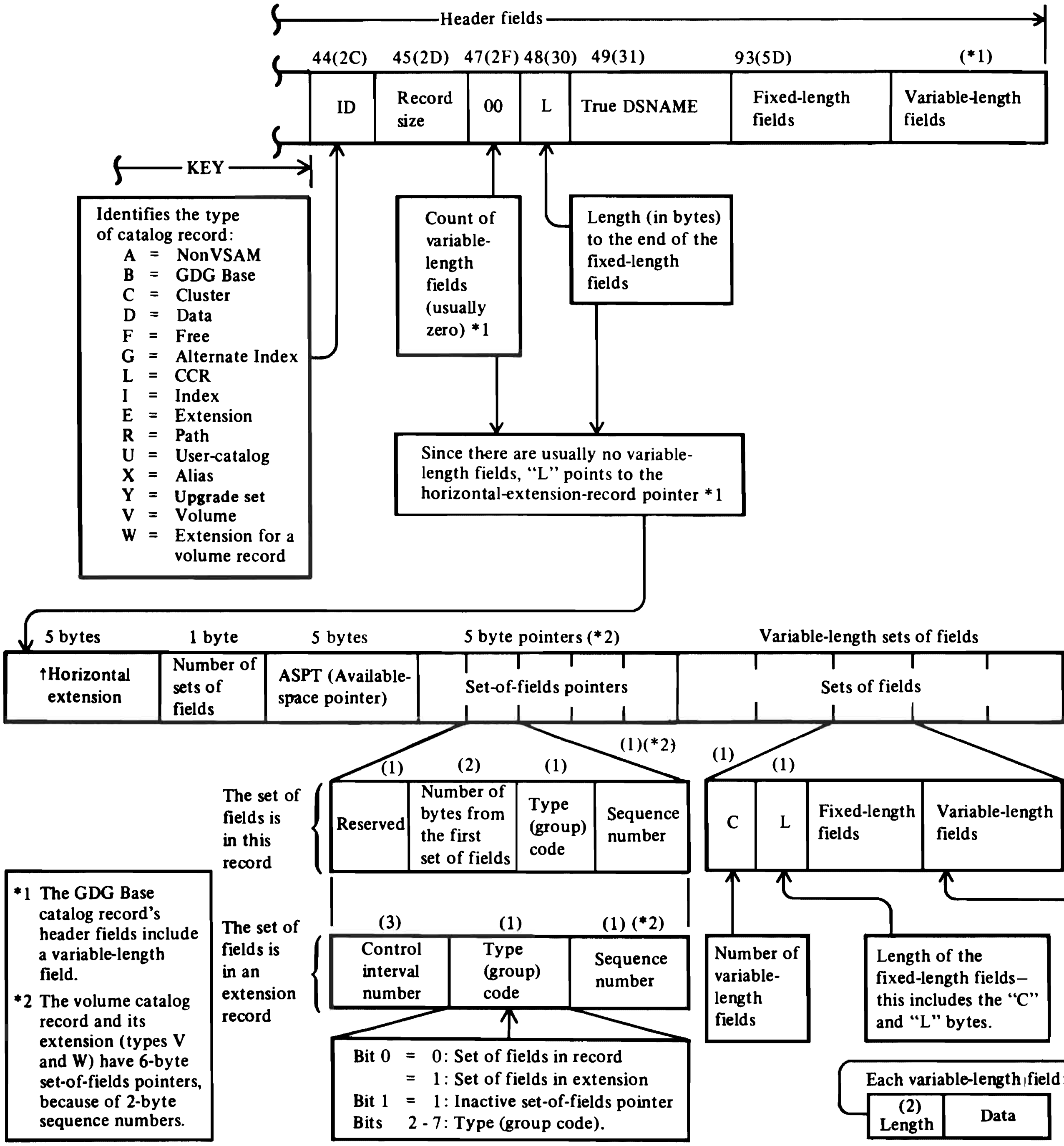


Figure 39 (Part 2 of 2). Catalog Record—General Format

Sets of Fields in the Catalog Records

Related fields of information are grouped into sets of fields so they can be treated as a unit. For example, all fields relating to one volume on which a data set resides are grouped together. If a data set resides on three volumes, there are three sets of volume information. All pointers to volume information sets of fields are contiguous, however. These sets of volume information fields are not necessarily contiguous. Each pointer to a set of fields contains a code that identifies the kind of information it contains so that the sets of fields that contain volume information can be distinguished from other sets of fields. Note that it is possible for one record to contain many sets of fields.

Following are the sets of fields that can occur in Cluster, Alternate Index, Path, Upgrade, Data, Index, Alias, GDG Base, NonVSAM, and User-Catalog records:

- AMDSB (Access Method Data Statistics Block), which appears in Data and Index records. Only one copy of an AMDSB appears in a record. A pointer to AMDSB information contains a code of 1.
- Association information, which appears in Data, Index, Cluster, Alternate Index, Path, Upgrade, User Catalog, NonVSAM, Alias, and GDG Base catalog records. Figure 40 illustrates the associations that can occur in these records. Each arrow represents an association. Associations shown by a broken line exist only when the base cluster is a key-sequenced data set. A cluster might not have any alternate indexes, paths, upgrade sets, or alias paths; a cluster's alternate index might not be part of the cluster's upgrade set. Note that multiple alternate indexes, paths, nonVSAMs, aliases, and alias paths may exist.
- Volume information, which appears in Data, Index, User Catalog and NonVSAM records. This set of fields describes all of the direct-access device space allocated to the data set (or index, etc.) on a particular volume. A separate set of volume information fields is used to describe the space on each volume. If the data set's space on a volume is divided into key ranges, each key range is described in a separate set of volume information fields. As many sets of volume information fields as are required to describe allocated space can appear. A pointer to volume information contains a code of 3.
- Password information, which can appear in Data, Index, Alternate Index, Path, and Cluster records. This set of fields contains the security information for a data set (or index, etc.). Only one set of password information fields can appear. A pointer to password information contains a code of 4.

The Volume record can also contain sets of fields, as follows:

- Track allocation information (Space Map set of fields). This set of fields describes each track on the volume as allocated to a VSAM object or unallocated. Each volume record contains as many of these sets of fields as are required to describe the entire volume. A pointer to track allocation information contains a code of 5.
- VSAM data space information (Data Space Group set of fields). This set of fields describes a VSAM data space on the volume. One set of fields is required to describe each data space and its extents on the volume. A pointer to data space information contains a code of 6.

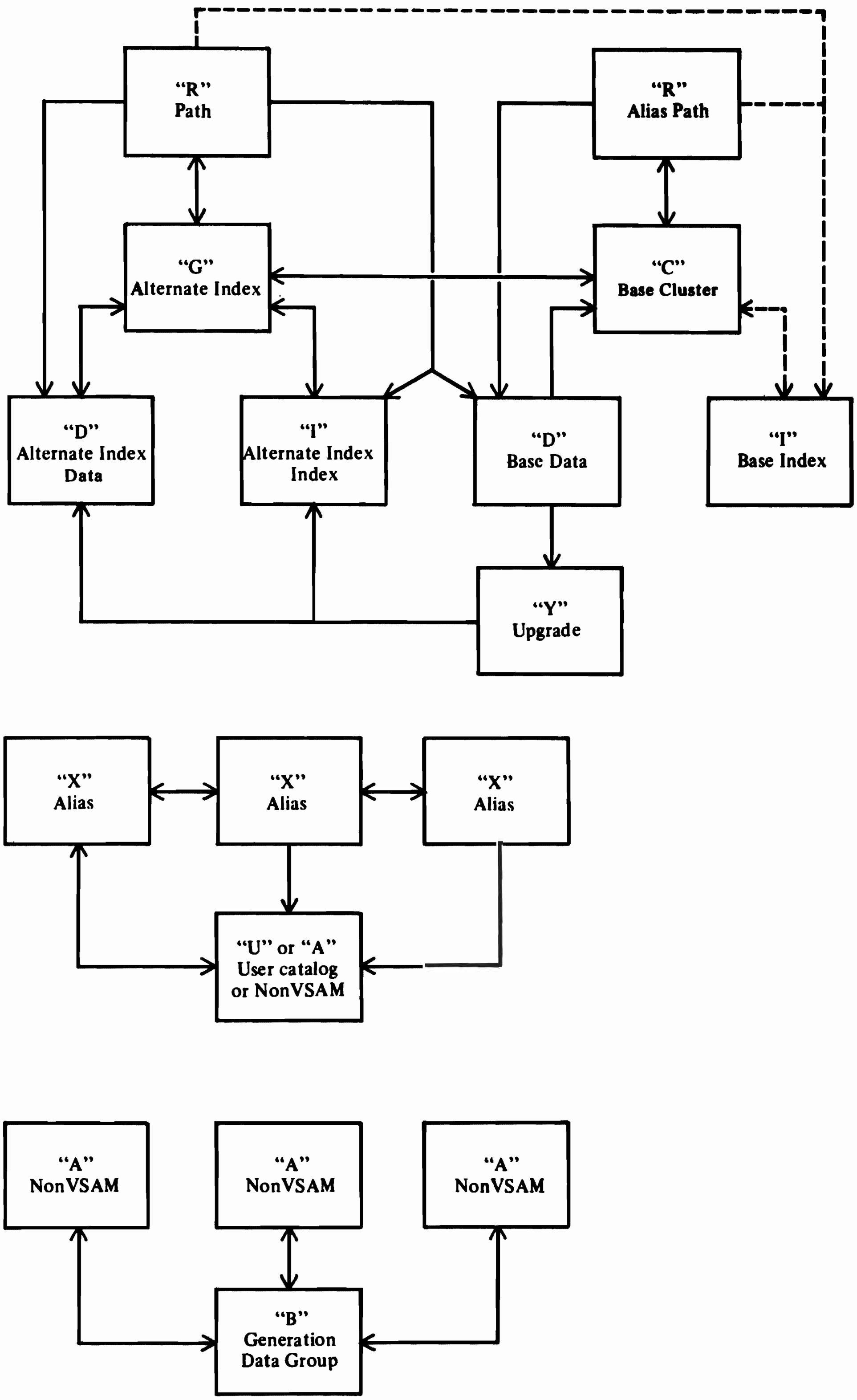


Figure 40. Catalog Record Associations

- Data set information (Data Set Directory Entry set of fields). This set of fields describes a data set that resides in a VSAM data space. One set of fields is required for each data set. A pointer to data set information contains a code of 8.

Note: If a Cluster, Alternate Index, Upgrade, Data, Index, NonVSAM, User Catalog, GDG Base, or Volume record is extended, these sets of fields (except for the AMDSB set of fields) are moved, as required, into an Extension record.

Locating Fields in Catalog Records

A field name dictionary, which is part of the catalog management code, allows catalog management to locate fields within catalog records by name. The dictionary also allows for combination field names—each combination field name allows catalog management to locate a group of related fields.

Catalog records and the field-name dictionary are described in the topics that follow.

Catalog Recovery Area Record Descriptions

Catalogs that are defined with the recoverable attribute are associated with one or more CRAs (catalog recovery areas). A CRA is a VSAM entry-sequenced data set, and every volume owned by a recoverable catalog contains one CRA.

Each CRA contains three types of 512-byte records:

- Self-describing records
- Duplicate copies of VSAM catalog entry records
- CRA free records

Self-describing records and free records occupy control intervals 0-8 in the CRA. The control intervals and the specific record they contain are:

CI Number	Record Type	Description
0	D	Data set record, which describes the CRA data component of the CRA cluster
1	F	Free record
2	C	Cluster record, which describes the CRA cluster
3	L	Control record, which manages control interval allocation in the CRA
4	F	Free record
5	E	Extension record, which is an extension of the data set record in control interval 0
6,7,8	F	Free records

The formats and contents of the self-describing records are shown later in the "Data Areas" section.

Duplicate copies of catalog records are recorded in the CRA in control intervals 9-n. The volume record for the CRA volume is in control interval 9. The format and content of these duplicated records are identical to their counterparts in the recoverable catalog associated with the CRA.

Each catalog record that is not self-describing is duplicated in a specific CRA. The table that follows shows which CRA contains a given catalog record. In the table, *initial volume* is the first volume on which space was allocated for the entity.

Catalog Entry	Record Type	CRA
Volume records	V,W	Subject volume
KSDS cluster records	C,E	Initial prime index volume
KSDS data records	D,E	Initial prime index volume
KSDS index records	I,E	Initial prime index volume
AIX cluster record (KSDS)	G,E	Initial prime index volume of the base cluster
AIX data record (KSDS)	D,E	Initial prime index volume of the base cluster
AIX index record (KSDS)	I,E	Initial prime index volume of the base cluster
AIX cluster records (ESDS)	G,E	Initial base data volume
AIX data records (ESDS)	D,E	Initial base data volume
AIX index records (ESDS)	I,E	Initial base data volume
Path records (KSDS, no AIX)	R,E	Initial prime index volume
Path records (ESDS, no AIX)	R,E	Initial base data volume
Path records (KSDS, AIX)	R,E	Initial prime index volume of the base cluster
Path records (ESDS, AIX)	R,E	Initial base data volume
Upgrade records (KSDS)	Y,E	Initial prime index volume of the base cluster
Upgrade records (ESDS)	Y,E	Initial base data volume
NonVSAM, Generation Data Group, User catalog, and Alias records	A,B,E,U,X	Catalog volume

True-Name Catalog Record Format

The True Name catalog record associates the volume serial number, data set DSNAME, alternate (alias) DSNAME of a nonVSAM data set or user catalog, user catalog DSNAME, or cluster, alternate index, or path DSNAME specified by the user with the control interval number of the catalog record that describes the object (volume, data set, alias, user catalog, alternate index, path, or cluster). True Name records are contained in the high-address part of the catalog and are pointed to by the catalog's index records. The True Name record is retrieved using key-sequenced processing. The catalog management modules convert the control interval number in the True Name record to an RBA for entry-sequenced processing.

True Name records are 47 bytes long; several might be contained in a catalog's (512-byte) control interval.

Offset	Bytes and Bit Pattern	Description
0 (0)	44	DSNAME of a data set, cluster, user catalog, nonVSAM data set, alternate index, path, or alternate DSNAME (alias), filled on the right with blanks, or a volume serial number, filled on the right with zeros, specified by the user.
44 (2C)	3	Control interval number of the catalog record that describes the object.

Catalog Control Record (CCR) Format

The catalog control record (CCR) is used by catalog management to control the allocation of control intervals in the low-address part of the catalog, where catalog records, excluding the True Name records and the index, reside. The CCR also shows the catalog's high-used and high-allocated RBA values. The catalog control record is the fourth record (control interval) in the catalog.

For a request of one catalog record, catalog management tries to use a record that was freed because of deletion. This process is done before using unassigned space. If more than one catalog record is needed, catalog management tries to use contiguous unassigned space in the current extent; if sufficient unassigned space is not available, records that have been deleted are used.

Offset	Bytes and Bit Pattern	Description	
0 (0)	44	Key.	
		Byte	**Meaning**
		0	Zeros.
		1-3	Control interval number of this record.
		4	Release indicator.
		5-43	Zeros.
44 (2C)	1	Record type—"L."	
45 (2D)	3	Number of the highest control interval within the current extent.[1]	
48 (30)	3	Number of the next free control interval of those that have not been previously assigned.[1]	
51 (33)	3	Count of deleted control intervals, that is, the count of control intervals that are free because of deletion.[2]	

Catalog Control Record Format (continued)

Offset	Bytes and Bit Pattern	Description
54 (36)	3	First deleted control interval in a chain of control intervals that are free because of deletion.[2]

The following fields are used to keep track of the RBA values that denote the current logical end (high RBA) of parts of the catalog:

Offset	Bytes and Bit Pattern	Description
57 (39)	4	Data, low key range: high-key RBA
61 (3D)	4	Data, low key range: high-used RBA
65 (41)	4	Data, low key range: high-allocated RBA
69 (45)	4	Data, high key range: high-key RBA
73 (49)	4	Data, high key range: high-used RBA
77 (4D)	4	Data, high key range: high-allocated RBA
81 (51)	4	Index, high level: high-used RBA
85 (55)	4	Index, high level: high-allocated RBA
89 (59)	4	Index, low key range—sequence set: high-used RBA
93 (5D)	4	Index, low key range—sequence set: high-allocated RBA
97 (61)	4	Index, high key range—sequence set: high-used RBA
101 (65)	4	Index, high key range—sequence set: high-allocated RBA

1 This field is used to keep track of unassigned space within the current extent.
2 This field is used to keep track of previously-used records that are now available for use as other catalog records.

Free Catalog Record Format

The Free catalog record indicates that the control interval in which it resides is free and points to the next control interval that is free because of deletion. Note that the Free catalog record is used only to mark a record that was used and has been deleted. The free space (control intervals) in the catalog that has never been assigned is not represented by Free catalog records.

Offset	Bytes and Bit Pattern	Description	
0 (0)	44	Key.	
		Byte	**Meaning**
		0	Zeros.
		1-3	Control interval number of this record.
		4-43	Zeros.
44 (2C)	1	Record type—"F."	
45 (2D)	3	Control interval number of the next free control interval.	

Data and Index Catalog Record Format

Data and Index records describe data sets and their indexes. A Data record can also describe a pagespace, which is similar to an entry-sequenced VSAM data set.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial.[1]
11 (B)	3	CRAIDNO	CRA control interval number.[1]
14 (E)	4	CRADEVT	CRA device type.[1]
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	4	CRADITS	Time stamp. Data/index identifier.
26 (1A)	1	SECFLAGS	Security attributes.
	1...		Indicates record is RACF-protected.
	.xxx xxxx		Reserved.
27 (1B)	17		Zeros.
44 (2C)	1	ENTYPE	Record type—"D" for a Data record or "I" for an Index record.
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an Extension record. Always zero for a Data and Index catalog record.
48 (30)	1		Length of the fixed-length fields in this record, excluding any fixed-length fields following displacement 143 (X'8F'). This value is always equal to the displacement from the beginning of the record to the pointer to an Extension record.
49 (31)	44	ENTNAME	For a Data or Index catalog record, the data set's DSNAME.
93 (5D)	8	OWNERID	Owner of the data set, specified when the data set was defined.
101 (65)	3	DSETCRDT	Data set creation date, in packed-decimal form YDD, specified when the data set was defined.
104 (68)	3	DSETEXDT	Data set expiration date, in packed-decimal form YDD, specified when the data set was defined.

Data and Index Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
107 (6B)	1	ATTR1	Data set attributes, which are defined in Access Method Services commands, as follows:
	1...		Speed, which indicates that direct-access device storage for the data set or index is not to be preformatted before records are written.
	.1..		Unique, which indicates that this data set or index must reside in a data space all its own.
	..1.		The cluster associated with this component is reusable.
	...1		Erase, which indicates that the data set or index is to be overwritten with binary zeros when deleted.
	.... 1...		This catalog is recoverable.
	.... .1..		Inhibit update, which indicates that the data set or index is not to be updated.
	.... ..1.		Temporary export, which indicates that the original copy of this data set or index is not to be deleted, even though another copy of it exists somewhere else.
	1		Track overflow—can be set on only in a Data catalog record that describes a pagespace.
108 (6C)	1	ATTR2	Data set sharing attributes as follows:
	00..		The data set can be shared by READ users *or* it can be used by one UPDATE/OUTPUT user.
	01..		The data set can be shared by READ users *and* one UPDATE/OUTPUT user.
	10..		The data set can be fully shared.
	11..		The data set can be fully shared, with assistance supplied by VSAM.
			Data set sharing attributes across systems, as follows:
	..00		Reserved.
	..01		Reserved.
	..10		The data set can be fully shared.
	..11		The data set can be fully shared, with assistance supplied by VSAM.
	.... ..1.		Internal system data set
	1		Component is not usable.
	.... xx..		Reserved.
109 (6D)	1	OPENIND	Open indicator flag; if this byte contains X‘80’, the data set is open for output.
110 (6E)	4	BUFSIZE	Minimum buffer size.
114 (72)	3	PRIMSPAC	Primary space allocation for the data set or index, specified when the data set or index was defined.
117 (75)	3	SCONSPAC	Secondary space allocation for the data set or index, specified when the data set or index was defined.

Data and Index Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
120 (78)	1	SPACOPTN	Space options flags.
	10..		Track request, which indicates that space allocation was specified in tracks.
	11..		Cylinder request, which indicates that space allocation was specified in cylinders.
	..xx xxxx		Reserved.
121 (79)	4	HURBADS	High-used RBA of the data set or index.
125 (7D)	4	HARBADS	High-allocated RBA of the data set or index.
129 (81)	4	LRECL	For a Data catalog record, the logical record size of the data set; for an Index catalog record, always X 'FF's.
133 (85)	2	USERINFO	User information for the DOS/VS indexed-sequential access method compatibility interface.
135 (87)	8	EXCPEXIT	Exception exit.

The following six-byte entry contains control information for the sets of fields that follow it.

Offset	Bytes and Bit Pattern	Field Name	Description
143 (8F)	5		Pointer to the horizontal Extension record. If this record is not continued in an Extension record, this field contains zeros.
148 (94)	1		The number of set-of-fields pointers that follow.[2]
			Note: The first set-of-fields pointer contains sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.
149 (95)	VL		5-byte pointers to sets of fields within the record.

Byte	Meaning
0	Reserved.
1-2	Displacement of the set of fields from the beginning of all sets of fields in this record.
3	Bits 0 and 1 are set to zero. If bit 1 is on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 of this byte and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to.
4	Sequence number of the set of fields pointed to, by type (group) code. For example, all sets of fields associated with a type (group) code of 2 are in one sequence.

Data and Index Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
	VL		5-byte pointers to sets of fields contained in vertical Extension records. **Byte** **Meaning** 0-2 Control interval number of the Extension record that contains this set of fields. 3 Bits 0 and 1 are set to B'10'. Bits 2 through 7 contain a type (group) code that describes the set of fields pointed to. 4 Sequence number of the set of fields pointed to, by type (group) code. For example, all sets of fields associated with a type (group) code of 2 are in one sequence.

1 Zeros if the catalog is not recoverable or if there is no associated CRA volume.

2 Fields describing (a) the AMDSB, (b) the control interval number of a Cluster record associated with this record, (c) the volumes on which a data set or index resides, and (d) the password information associated with a data set are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in each set of fields.

AMDSB (Access Method Data Statistics Block) Set of Fields Format

The AMDSB set of fields contains a copy of the AMDSB control block, and is updated each time the data set is closed. This set of fields is associated with a pointer that contains a type (group) code of 1.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	96	AMDSBCAT	Copy of the AMDSB—Access Method Data Statistics Block. See "Data Areas" for detailed information about the AMDSB.

Association (Cluster) Set of Fields Format

The control interval number of the cluster catalog record associated with the data set or index catalog record is contained in the association set of fields. This set of fields is associated with a pointer that contains a type (group) code of 2.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	1	TYPE	"C," which indicates that this record is associated with a Cluster record, or "G," which indicates this is the data component or the index component of an alternate index. The data component of a cluster may have an additional "Y" association if the cluster has an alternate index that is part of the upgrade set.
3 (3)	3	NAME	Control interval number of the Cluster or Alternate Index record associated with this record.

Volume Information Set of Fields Format

All extents allocated to the data set, index, or data set's key range on a volume are described by a volume information set of fields. This set of fields is associated with a pointer that contains a type (group) code of 3.

Offset	Byte and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	4	DEVTYP	Device type.
6 (6)	6	VOLSER	Volume serial number.
12 (C)	2	FILESEQ	File sequence number. (This field is provided for compatibility with the OS/VS catalog, and is used for nonVSAM data sets that reside on tape volumes.)

Data Set and Index Catalog Record: Volume Information Set of Fields Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
14 (E)	1	VOLFLG	Volume flags, as follows:
	1...		Prime, which indicates that this volume was allocated when the data set was defined or that a data set that is not divided into parts according to key (a key-range data set) has been extended to this volume.
	.1..		Candidate, which indicates that this volume is available for use by the data set described by this record.
	..1.		Overflow, which indicates that this volume is being used by a data set that is divided into parts according to key (a key-range data set), but this volume was not allocated when the data set was defined.
	...x xxxx		Reserved
15 (F)	1	NOEXTNT	Number of extents allocated in this set of extents on this volume for this data set.
16 (10)	4	HKRBA	RBA of the data control interval with the high key.
20 (14)	4	HURBA	High used RBA.
24 (18)	4	HARBA	High allocated RBA.
28 (1C)	4	PHYBLKSZ	Block size.
32 (20)	2	NOBLKTRK	Number of blocks per track.
34 (22)	2	NOTRKAU	Number of tracks per allocation unit.
36 (24)	1	ITYPEXT	Flags:
	1...		In an index record: the sequence set is with the data.
	.1..		The extents are not preformatted
	..xx xxxx		Reserved
37 (25)	2	DSDIRSN	Data set directory sequence number in the volume record.
39 (27)	4	NOBYTTRK	Number of bytes per track—nonexistent unless the record describes a pagespace.
43 (2B)	4	NOBYTAU	Number of bytes per allocation unit—nonexistent unless the record describes a pagespace.

Data Set and Index Catalog Record: Volume Information Set of Fields Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
47 (2F)[1]	VL	LOKEYV	Low key on the volume. This field can be a maximum of 64 bytes long; the first two bytes (control information) indicate the length of the field.
	VL	HIKEYV	High key on the volume. This field can be a maximum of 64 bytes long; the first two bytes (control information) indicate the length of the field.
	VL	EXTENT	This field contains a 2-byte length field (control information), followed by a 20-byte field for each extent. The 20-byte field describes the start and end of the extent, in the form SSCCHHCCHHTTDDDDDDDD, where SS is the data space extent's sequence number, CCHHCCHH is the low and high cylinder and head, TT is the number of tracks, and DDDDDDDD is the low and high RBA of the extent.

[1] The offset to LOKEYV is 39 (27) when the record doesn't describe a pagespace (and NOBYTTRK and NOBYTAU are nonexistent). The second byte of the record contains the displacement.

Password Set of Fields Format

Password information, if any, is contained in the password set of fields. This set of fields is associated with a pointer that contains a type (group) code of 4.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	32	PASSWORD	Four eight-character passwords, in the following order: MASTER, CONTROL INTERVAL, UPDATE, and READ-ONLY.
36 (24)	8	PASSPRMT	Password prompting code name that allows the operator to provide the correct password without displaying the data set's DSNAME.
42 (2A)	2	PASSATMP	Maximum number of attempts allowed for the operator or TSO operator to provide the correct password.
44 (2C)	8	USVRMDUL	Name of user's security-verification module, if any.
52 (34)	VL	USERAREC	User-authorization record. This field can be a maximum of 256 bytes long.

Cluster Catalog Record Format

The Cluster catalog record points to its data set's Data catalog record. If the data set is key-sequenced, the Cluster catalog record also points to the data set's Index catalog record. The Cluster catalog record can point to a Data catalog record that describes a pagespace (conceptually an entry-sequenced VSAM data set). In addition, the cluster may point to one or more alternate indexes or one or more alias paths.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial.[1]
11 (B)	3	CRAIDNO	CRA control interval number.[1]
14 (E)	4	CRADEVT	CRA device type.[1]
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	4		Zeros.
26 (1A)	1	SECFLAGS	Security attributes.
	1...		Indicates record is RACF-protected.
	.xxx xxxx		Reserved
27 (1B)	17		Zeros
44 (2C)	1	ENTYPE	Record type—"C."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an Extension record. Always zero for a Cluster catalog record.
48 (30)	1		Length of the fixed-length fields in this record, excluding any fixed-length fields that follow displacement 108 (X'6C'). This value is always equal to the displacement from the beginning of the record to the pointer to an Extension record.
49 (31)	44	ENTNAME	DSNAME of the cluster described by this record.
93 (5D)	8	OWNERID	Owner of the cluster, specified when the cluster was defined.
101 (65)	3	DSETCRDT	Cluster creation date, in packed-decimal form YDD, specified when the cluster was defined.
104 (68)	3	DSETEXDT	Cluster expiration date, in packed-decimal form YDD, specified when the cluster was defined.
107 (6B)	1	CATTR	Cluster attributes:
	.... ..1.		The cluster describes a swapspace
	1		The cluster describes a pagespace
	xxxx xx..		Reserved

Cluster Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
The following 6-byte entry contains control information for the sets of fields that follow it.			
108 (6C)	5		Pointer to the horizontal Extension record. If this record is not continued on an Extension record, this field contains zeros.
113 (71)	1		The number of set-of-fields pointers that follow.[2] **Note:** The first set-of-fields pointer might be special (meaning that the field-name dictionary permits catalog management to locate information that is not contained in catalog records); if so, its sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.
114 (72)	VL		5-byte pointers to sets of fields within the record. **Byte** **Meaning** 0 Reserved. 1-2 Displacement of the set of fields from the beginning of all sets of fields in this record. 3 Bits 0 and 1 are set to zero. If bit 1 is on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 of this byte and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to. 4 Sequence number of the set of fields pointed to by code. For example, all sets of fields associated with a code of 2 are in one sequence.
	VL		5-byte pointers to sets of fields contained in vertical Extension records.

Cluster Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
			Byte **Meaning**
			0-2 Control interval number of the Extension record that contains this set of fields.
			3 Bits 0 and 1 are set to B'10'. Bits 2 through 7 contain a type (group) code that describes set of fields pointed to.
			4 Sequence number of the set of fields pointed to, by type (group) code. For example, all sets of fields associated with a code of 2 are in one sequence.

[1] Zeros if the catalog is not recoverable or if there is no associated CRA entry.
[2] Fields describing (a) the control interval number of a Data or Index catalog record associated with this cluster, or (b) the password information associated with a data set are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in each set of fields.

Association (Data and Index) Set of Fields Format

The control interval number of the Data and Index catalog record associated with the cluster is contained in an association set of fields. This set of fields is associated with a pointer that contains a type (group) code of 2.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information.
			Byte **Meaning**
			0 Count of the number of variable-length fields in this set of fields.
			1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	1	TYPE	If this entry describes an Index catalog record, "I;" if this entry describes a Data catalog record, "D;" if this entry describes an alternate index entry, "G;" if this entry describes an alias path entry, "R."
3 (3)	3	NAME	Control interval number of a Data, Index, Alternate Index, or Alias Path catalog record that describes part of the cluster described by this record.

Password Set of Fields Format

Password information, if any, is contained in the password set of fields. This set of fields is associated with a pointer that contains a type (group) code of 4.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	32	PASSWORD	Four eight-character passwords, in the following order: MASTER, CONTROL INTERVAL, UPDATE, and READ-ONLY.
34 (22)	8	PASSPRMT	Password prompting code name that allows the operator to provide the correct password without displaying the cluster's DSNAME.
42 (2A)	2	PASSATMP	Maximum number of attempts allowed for the operator or TSO operator to provide the correct password.
44 (2C)	8	USVRMDUL	Name of the user's security-verification module, if any.
52 (34)	VL	USERAREC	User-authorization record. This field can be a maximum of 256 bytes long.

Alternate Index Catalog Record Format

The alternate index record describes the data and index components associated with the alternate index. In addition, it points to the related cluster entry and it can point to one or more path entries. The alternate index grouping is similar to the grouping of a key-sequenced data set except for different record types ("G" rather than "C").

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial.
11 (B)	3	CRAIDNO	CRA control interval number.
14 (E)	4	CRADEVT	CRA device type.
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	4		Zeros.
26 (1A)	1	SECFLAGS	Security attributes.
	1...		Indicates record is RACF-protected.
	.xxx xxxx		Reserved
27 (1B)	17		Zeros.

Alternate Index Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
44 (2C)	1	ENTYPE	Record type—"G."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an extension record (always zero).
48 (30)	1		Length of nonrepeating fixed-length fields (always X'6C').
49 (31)	44	ENTNAME	Name of the alternate index described by this record.
93 (5D)	8	OWNERID	Owner of the alternate index described by this record; specified when the alternate index was defined.
101 (65)	3	DSETCRDT	Alternate index creation date, in the packed-decimal form YDD.
104 (68)	3	DSETEXDT	Alternate index expiration date, in the packed-decimal form YDD.
107 (6B)	1	RGATTR	Alternate index attributes:
	1...		If there is an association in the upgrade entry, indicates that this alternate index is a member of the upgrade set.
	.xxx xxxx		Reserved.

The following 6-byte entry contains control information for the sets of fields that follow it.

Offset	Bytes and Bit Pattern	Field Name	Description
108 (6C)	5		Pointer to the horizontal extension record. If this record is not continued on an extension record, this field contains zeros.
113 (71)	1		The number of-set-of fields pointers that follow.[1] **Note:** The first set-of-fields pointer might be special (meaning that the field-name dictionary permits catalog management to locate information that is not contained in catalog records); if so its sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.

Alternate Index Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
114 (72)	VL		5-byte pointers to sets of fields within the record. **Byte** **Meaning** 0 Reserved. 1-2 Displacement of the set of fields from the beginning of all sets of fields in this record. 3 Bits 0 and 1 are set to zero. If bit 1 is on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 of this byte and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to. 4 Sequence number of the set of fields pointed to by code. For example, all sets of fields associated with a code of 2 are in one sequence.
	VL		5-byte pointers to sets of fields contained in vertical extension records. **Byte** **Meaning** 0-2 Control interval number of the extension record that contains this set of fields. 3 Bits 0 and 1 are set to B'10'. Bits 2 through 7 contain a type (group) code that describes set of fields pointed to. 4 Sequence number of the set of fields pointed to, by type (group) code. For example, all sets of fields associated with a code of 2 are in one sequence.

[1] Fields describing (a) the control interval number of a Data or Index catalog record associated with this cluster, or (b) the password information associated with a data set are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in each set of fields.

Association Set of Fields Format

The associations in this entry are partially ordered; however, no assumptions should be made as to the relative placement or physical position of these associations in the alternate index record.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information.
2 (2)	1	TYPE	Record type pointed to by the following control interval number.
3 (3)	3	NAME	Control interval number.

The association ordering by record type is:

Record Type	Description
D	Alternate index data (D) component association with occurrence sequence number = 1
I	Alternate index index (I) component association with occurrence sequence number = 2
C	Base cluster (C) component association with occurrence sequence number = 3
R	Maximum of 252 path (R) associations

Password Set of Fields Format

Password information, if any, is contained in the password set of fields. This set of fields is associated with a pointer that contains a type (group) code of 4.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	32	PASSWORD	Four eight-character passwords, in the following order: MASTER, CONTROL INTERVAL, UPDATE, and READ-ONLY.
34 (22)	8	PASSPRMT	Password prompting code name that allows the operator to provide the correct password without displaying the cluster's DSNAME.
42 (2A)	2	PASSATMP	Maximum number of attempts allowed for the operator or TSO operator to provide the correct password.
44 (2C)	8	USVRMDUL	Name of the user's security-verification module, if any.
52 (34)	VL	USERAREC	User-authorization record. This field can be a maximum of 256 bytes long.

Path Catalog Record Format

The path record describes an alternate index and its associated base data set to give an alternate, logical view of the base data set. It also may be used as an alias for a base data set to inhibit upgrade set unit allocation.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA (catalog recovery area) volume serial.
11 (B)	3	CRAIDNO	CRA control interval number.
14 (E)	4	CRADEVT	CRA device type.
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	4		Zeros.
26 (1A)	1	SECFLAGS	Security attributes.
	1...		Indicates record is RACF-protected.
	.xxx xxxx		Reserved.
27 (1B)	17		Zeros.
44 (2C)	1	ENTYPE	Record type—"R."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the address of an extension record (always zero).
48 (30)	1		Length of nonrepeating fixed-length fields (always X'6C').
49 (31)	44	ENTNAME	Name of the path described by this record.
93 (5D)	8	OWNERID	Owner of the path; specified when the path was defined.
101 (65)	3	DSETCRDT	Path creation date; in the packed-decimal form YDD.
104 (68)	3	DSETEXDT	Path expiration date; in the packed-decimal form YDD.
107 (6B)	1	RGATTR	Path attributes:
	1...		Include upgrade set during unit allocation and when opening this path.
	.xxx xxxx		Reserved.
The following 6-byte entry contains control information for the sets of fields that follow it.			
108 (6C)	5		Pointer to the horizontal extension record. If this record is not continued on an extension record, this field contains zeros.

Path Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
113 (71)	1		The number of set of fields pointers that follow.[1] **Note:** The first set-of-fields pointer might be special (meaning that the field-name dictionary permits catalog management to locate information that is not contained in catalog records); if so, its sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.
114 (72)	VL		5-byte pointers to sets of fields within the record.
			Byte / **Meaning**
			0 — Reserved.
			1-2 — Displacement of the set of fields from the beginning of all sets of fields in this record.
			3 — Bits 0 and 1 are set to zero. If bit 1 is on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 of this byte and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to.
			4 — Sequence number of the set of fields pointed to by code. For example, all sets of fields associated with a code of 2 are in one sequence.
		VL	5-byte pointers to sets of fields contained in vertical extension records.
			Byte / **Meaning**
			0-2 — Control interval number of the extension record that contains this set of fields.
			3 — Bits 0 and 1 are set to B'10'. Bits 2 through 7 contain a type (group) code that describes set of fields pointed to.
			4 — Sequence number of the set of fields pointed to, by type (group) code. For example, all sets of fields associated with a code of 2 are in one sequence.

[1] Fields describing (a) the control interval number of a Data or Index catalog record associated with this cluster, or (b) the password information associated with a data set are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in each set of fields.

Association Set of Fields Format

The associations in this entry are ordered in the sense that each association occurrence has a defined group occurrence sequence number; however, no assumptions should be made as to the relative placement or physical position of these associations in the path record.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information.
2 (2)	1	TYPE	Record type which is pointed to by the following control interval number.
3 (3)	3	NAME	Control interval number.

If this record describes a path over an alternate index, the association ordering by record type is:

Record Type	Description
G	Alternate index (G) entry association with occurrence sequence number = 1
D	Alternate index data (D) component association with occurrence sequence number = 2
I	Alternate index index (I) component association with occurrence sequence number = 3
D	Base data (D) component association with occurrence sequence number = 4
I	Base index (I) component association with occurrence sequence number = 5. This association exists only if the base cluster is a key-sequenced data set.

If this record describes a path over a base cluster, the association ordering by record type is:

Record Type	Description
C	Base cluster (C) component association with occurrence sequence number = 1
D	Base data (D) component association with occurrence sequence number = 2
I	Base index (I) component association with occurrence sequence number = 3. This association exists only if the base cluster is a key-sequenced data set.

Password Set of Fields Format

Password information, if any, is contained in the password set of fields. This set of fields is associated with a pointer that contains a type (group) code of 4.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	32	PASSWORD	Four eight-character passwords, in the following order: MASTER, CONTROL INTERVAL, UPDATE, and READ-ONLY.
34 (22)	8	PASSPRMT	Password prompting code name that allows the operator to provide the correct password without displaying the data set's DSNAME.
42 (2A)	2	PASSATMP	Maximum number of attempts allowed for the operator or TSO operator to provide the correct password.
44 (2C)	8	USVRMDUL	Name of the user's security-verification module, if any.
52 (34)	VL	USERAREC	User-authorization record. This field can be a maximum of 256 bytes long.

Upgrade Catalog Record Format

The upgrade record describes all the alternate indexes that make up the upgrade set. It is pointed to by an association in the base data component.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial number.
11 (B)	3	CRAIDNO	CRA control interval number.
14 (E)	4	CRADEVT	CRA device type.
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	22		Zeros.
44 (2C)	1	ENTYPE	Record type—"Y."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length records that precede the pointer to an extension record (always zero).
48 (30)	1		Length of nonrepeating fixed-length fields (always X'31').

Upgrade Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
The following 6-byte entry contains control information for the sets of fields that follow it.			
49 (31)	5		Pointer to the horizontal extension record. If this record is not continued on an extension record, this field contains zeros.
54 (36)	1		The number of set of fields pointers that follow.[1]
			Note: The first set-of-fields pointer contains sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.

Upgrade Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
55 (37)	VL		5-byte pointers to sets of fields within the record. **Byte** **Meaning** 0 Reserved. 1-2 Displacement of the set of fields from the beginning of all sets of fields in this record. 3 Bits 0 and 1 are set to zero. If bit 1 is on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 of this byte and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to. 4 Sequence number of the set of fields pointed to by code. For example, all sets of fields associated with a code of 2 are in one sequence.
	VL		5-byte pointers to sets of fields contained in vertical extension records. **Byte** **Meaning** 0-2 Control interval number of the extension record that contains this set of fields. 3 Bits 0 and 1 are set to B'10'. Bits 2 through 7 contain a type (group) code that describes set of fields pointed to. 4 Sequence number of the set of fields pointed to, by type (group) code. For example, all sets of fields associated with a code of 2 are in one sequence.

[1] Fields describing (a) the control interval number of a Data or Index catalog record associated with this cluster, or (b) the password information associated with a data set are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in each set of fields.

Association Set of Fields Format

The associations in this entry are an extension of the Release 1 format; they are actually twin associations consisting of a Release 1 data association and a Release 1 index association.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information.
2 (2)	1	TYPE	"D," which indicates the following control interval number is for the data component of an alternate index in the upgrade set.
3 (3)	3	NAME	Control interval number of the alternate index data component.
6 (6)	1	TYPE2	"I," which indicates the following control interval number is for the index component of an alternate index in the upgrade set.
7 (7)	3	NAME2	Control interval number of the alternate index index component.

These twin associations exist only in upgrade records (type "Y"), and the set of twin associations in any given upgrade record entry is always unique.

NonVSAM Catalog Record Format

The NonVSAM catalog record describes a data set organized differently from VSAM data set organization.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial.[1]
11 (B)	3	CRAIDNO	CRA control interval number.[1]
14 (E)	4	CRADEVT	CRA device type.[1]
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	22		Zeros.
44 (2C)	1	ENTYPE	Record type—"A."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an Extension record. Always zero for a NonVSAM catalog record.
48 (30)	1		Length of the fixed-length fields in this record, excluding any fixed-length fields following displacement 93 (X'5D'). This value is always equal to the displacement from the beginning of the record to the pointer to an Extension record.
49 (31)	44	ENTNAME	DSNAME of the nonVSAM data set described by this record.
93 (5D)	8	OWNERID	Owner of the data set; specified when the data set was defined.

NonVSAM Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
101 (65)	3	DSETCRDT	Data set creation date, in packed-decimal form YDD, the date the data set was defined.
104 (68)	3	DSETEXDT	Data set expiration date, in packed-decimal form YDD, specified when the data set was defined.

The following 6-byte entry contains control information for the set of fields that follow it.

Offset	Bytes and Bit Pattern	Field Name	Description
107 (6B)	5		Pointer to the horizontal Extension record. If this record is not continued on an Extension record, this field contains zeros.
112 (70)	1		The number of set of fields pointers that follow.[2]

Note: The first set-of-fields pointer contains sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.

Offset	Bytes and Bit Pattern	Field Name	Description
113 (71)	VL		5-byte pointers to sets of fields within the record.

Byte	Meaning
0	Reserved.
1-2	Displacement of the set of fields from the beginning of all sets of fields in this record.
3	Bits 0 and 1 are set to zero. If bit 1 is set on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to.
4	Sequence number of the set of fields pointed to.

NonVSAM Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
	VL		5-byte pointers to sets of fields contained in vertical extension records. **Byte** **Meaning** 0-2 Control interval number of the extension record that contains this set of fields. 3 Bits 0 and 1 are set to B'10'. Bits 2 through 7 contain a type (group) code that describes the set of fields pointed to. 4 Sequence number of the set of fields pointed to.

[1] Zeros if the catalog is not recoverable or if there is no associated CRA volume.
[2] Fields describing (a) the volumes on which the data set resides, (b) the first Alias record in the Alias record chain for the nonVSAM data set, or (c) if the nonVSAM data set is a generation data set, the GDG Base record are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in the set of fields.

Association (Alias) Set of Fields Format

The control interval number of the first alias catalog record in the alias record chain associated with the nonVSAM catalog record is contained in the association set of fields. This set of fields is associated with a pointer that contains a type (group) code of 2.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	1	TYPE	"X," which indicates that this record is associated with an Alias record.
3 (3)	3	NAME	Control interval number of the first Alias record in the Alias record chain associated with this record.

Association (GDG Base) Set of Fields Format

If the nonVSAM data set is a generation data set, it is associated with a generation data group. The control interval number of the generation data group (GDG) base catalog record is contained in the association set of fields. This set of fields is associated with a pointer that contains a type (group) code of 2.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	1	TYPE	"B," which indicates that this record is associated with a GDG Base record.
3 (3)	3	NAME	Control interval number of the GDG Base record associated with this record.

Volume Information Set of Fields Format

Each volume that contains space allocated to the nonVSAM data set is described by a volume information set of fields. This set of fields is associated with a pointer that contains a type (group) code of 3.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	4	DEVTYP	Device type.
6 (6)	6	VOLSER	Volume serial number.
12 (C)	2	FILESEQ	File sequence number. (This field is provided for compatibility with the OS/VS catalog, and is used for nonVSAM data sets that reside on tape volumes.)
14 (E)	1	VOLFLG	Volume flags, as follows:
	1...		Prime, which indicates that this volume was allocated when the data set described by this record was defined.
	..1.		Overflow, which indicates that this volume is being used by the data set, but this volume was not allocated when the data set was defined.
	.x.x xxxx		Reserved.
15 (F)	3	DSCBTTR	If the nonVSAM data set resides entirely on one volume, the DSCBTTR contains the location (TTR) of the data set's Format 1 DSCB in the volume's VTOC.

Alias Catalog Record Format

The alias catalog record allows a user to refer to a nonVSAM data set or user's catalog with an alternate DSNAME. If the data set or catalog has more than one alias, each alias is described in an alias catalog record. All alias catalog records associated with the data set or catalog are chained together.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial.[1]
11 (B)	3	CRAIDNO	CRA control interval number.[1]
14 (E)	4	CRADEVT	CRA device type.[1]
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	22		Zeros.
44 (2C)	1	ENTYPE	Record type—"X."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an Extension record. Always zero for an Alias catalog record.
48 (30)	1		Length of the fixed-length fields in this record, excluding any fixed-length fields following displacement 93 (X'5D').
49 (31)	44	ENTNAME	Alternate DSNAME (alias) of a nonVSAM data set or OS/VS2 user catalog.
The following 6-byte entry contains control information for the set of fields that follow it.			
93 (5D)	5		Pointer to the horizontal extension record. The Alias catalog record is never continued on an extension record, so this field contains zeros.
98 (62)	1		The number of set of fields pointers that follow.[2]

Alias Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
99 (63)	VL		5-byte pointers to sets of fields within the record. **Byte** **Meaning** 0 Reserved. 1-2 Displacement of the set of fields from the beginning of all sets of fields in this record. 3 Bits 0 and 1 are set to zero. If bit 1 is set on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to. 4 Sequence number of the set of fields pointed to.

[1] Zeros if the catalog is not recoverable or if there is no associated CRA volume.
[2] Fields describing (a) the NonVSAM or User-Catalog catalog record that describes the object for which this DSNAME is an alternate DSNAME (alias), or (b) previous and next Alias records in the Alias record chain are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in the set of fields.

Association (User-Catalog or NonVSAM) Set of Fields Format

The control interval number of the catalog record that contains the user catalog's or nonVSAM data set's real DSNAME is contained in the Alias catalog record's first association set of fields. This set of fields is associated with a pointer that contains a type (group) code of 2.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	1	TYPE	"U," which indicates that this record is associated with a User-Catalog record, or "A," which indicates that this record is associated with a NonVSAM record.
3 (3)	3	NAME	Control interval number of the User-Catalog or NonVSAM record associated with this record.

Association (Alias) Set of Fields Format

The control interval number of the next and previous alias catalog record in the alias record chain is contained in the association sets of fields, respectively. These sets of fields are associated with pointers that contain a type (group) code of 2.

If the Alias record is the first record in the Alias record chain, its second (next pointer) Association set of fields' NAME field contains zeros.

If the Alias record is the last record in the Alias record chain, its third (previous pointer) Association set of fields' NAME field contains zeros.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	1	TYPE	"X," which indicates that this record is associated with a Alias record.
3 (3)	3	NAME	Control interval number of the Alias record associated with this record.

Generation Data Group (GDG) Base Catalog Record Format

The GDG base catalog record allows a user to build and refer to generation data groups. The GDG base catalog record contains an association set of fields for each generation data set in the group.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial.[1]
11 (B)	3	CRAIDNO	CRA control interval number.[1]
14 (E)	4	CRADEVT	CRA device type.[1]
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	22		Zeros.
44 (2C)	1	ENTYPE	Record type—"B."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an Extension record. Always 1 for a GDG Base catalog record.
48 (30)	1		Length of the fixed-length fields in this record, excluding any fixed-length fields following displacement 109 (X'6D).

GDG Base Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
49 (31)	44	ENTNAME	DSNAME of the generation data group described by this record.
93 (5D)	8	OWNERID	Owner of the generation data group; specified when it was defined.
101 (65)	3	DSETCRDT	Creation date, in packed-decimal form YDD. The date the generation data group was defined.
104 (68)	3	DSETEXDT	Expiration date, in packed-decimal form YDD, specified when the generation data group was defined.
107 (6B)	1	GDGLIMIT	Maximum number of GDG levels
108 (6C)	1	GDGATTR	GDG attributes:
	x.......		Indicates how many data sets to delete from the catalog when GDGLIMIT is exceeded: 0-delete oldest one; 1-delete all.
	.x......		Indicates whether to scratch the deleted data set(s)'s format-1 DSCB: 0-don't scratch; 1-scratch.
	..xxxxxx		Reserved.
109 (6D)	VL	GENLVLS	Generation level difference string
The following 6-byte entry contains control information for the set of fields that follows it.			
	5[1]		Pointer to the horizontal Extension record. If this record is not continued on an Extension record, this field contains zeros.
	1		The number of set of fields pointers that follow.[2]
			Note: The first set-of-fields pointer contains sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.

GDG Base Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
	VL		5-byte pointers to sets of fields within the record. **Byte** **Meaning** 0 Reserved. 1-2 Displacement of the set of fields from the beginning of all sets of fields in this record. 3 Bits 0 and 1 are set to zero. If bit 1 is set on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to. 4 Sequence number of the set of fields pointed to.
	VL		5-byte pointers to sets of fields contained in vertical extension records. **Byte** **Meaning** 0-2 Control interval number of the extension record that contains this set of fields. 3 Bits 0 and 1 are set to B'10'. Bits 2 through 7 contain a type (group) code that describes the set of fields pointed to. 4 Sequence number of the set of fields pointed to.

1 Zeros if the catalog is not recoverable or if there is no associated CRA entry.

2 Fields describing each generation (nonVSAM) data set in the generation data group (GDG) are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in the set of fields.

Association (NonVSAM) Set of Fields Format

The control interval number of the nonVSAM catalog record that describes one of the generation data sets in the generation data group is contained in the association set of fields. This set of fields is associated with a pointer that contains a type (group) code of 2.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	1	TYPE	"A," which indicates that this record is associated with a NonVSAM record.
3 (3)	3	NAME	Control interval number of the NonVSAM record associated with this record.
6 (6)	4	GENLEVEL	Generation level—the 'xxxx' field in the generation data set DSNAME "dsname.gxxxxvyy".

User-Catalog Catalog Record Format

The User-Catalog catalog record describes a user catalog.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial.[1]
11 (B)	3	CRAIDNO	CRA control interval number.[1]
14 (E)	4	CRADEVT	CRA device type.[1]
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	22		Zeros.
44 (2C)	1	ENTYPE	Record type—"U."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an Extension record. Always zero for a User Catalog catalog record.
48 (30)	1		Length of the fixed-length fields in this record, excluding any fixed-length fields following displacement 93 (X'5D'). This value is always equal to the displacement from the beginning of the record to the pointer to an Extension record.
49 (31)	44	ENTNAME	DSNAME of the user catalog described by this record.

User-Catalog Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
The following 6-byte entry contains control information for the sets of fields that follow it.			
93 (5D)	5		Pointer to the horizontal Extension record. The User Catalog catalog record record is not continued on an Extension record, so this field contains zeros.
98 (62)	1		The number of set-of-fields pointers that follow.[2] **Note:** The first set-of-fields pointer contains sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.
99 (63)	VL		5-byte pointers to sets of fields within the record. **Byte** **Meaning** 0 Reserved 1-2 Displacement of the set of fields from the beginning of all sets of fields in this record. 3 Bits 0 and 1 are set to zero. If bit 1 is on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 of this byte and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to. 4 Sequence number of the set of fields pointed to.

[1] Zeros if the catalog is not recoverable or if there is no associated CRA volume.
[2] Fields describing (a) the volume on which the user's catalog resides or, (b) the first alias in the Alias record chain for a user's catalog are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in the set of fields.

Association (Alias) Set of Fields Format

The control interval number of the first Alias record in the Alias catalog record chain associated with the user-catalog catalog record is contained in the association set of fields. This set of fields is associated with a pointer that contains a type (group) code of 2.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	1	TYPE	"X," which indicates that this record is associated with a Alias record.
3 (3)	3	NAME	Control interval number of the first alias record in the alias record chain associated with htis record.

Volume Information Set of Fields Format

Each volume that contains space allocated to the user catalog is described by a volume information set of fields. This set of fields is associated with a pointer that contains a type (group) code of 3.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields.
2 (2)	4	DEVTYP	Device type.
6 (6)	6	VOLSER	Volume serial number.

Volume Catalog Record Format

The Volume record describes VSAM data spaces, their extents, and the data sets that reside in VSAM data spaces.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial.[1]
11 (B)	3	CRAIDNO	CRA control interval number.[1]
14 (E)	4	CRADEVT	CRA device type.[1]
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	22		Zeros.
44 (2C)	1	ENTYPE	Record type—"V."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an Extension record. Always zero for a Volume catalog record.
48 (30)	1		Length of the fixed-length fields in this record, excluding any fixed-length fields following displacement 127 (X'7F'). This value is always equal to the displacement from the beginning of the record to the pointer to an Extension record.
49 (31)	44	ENTNAME	8-byte volume serial number, filled with 36 bytes of binary zeros on the right, of the volume described by this record.
93 (5D)	8	VOLSTMP	Volume time stamp, which indicates when the first VSAM data space was defined on this volume.

Volume Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
101 (65)	20	VOLDVCHR	Device characteristics. **Byte** **Meaning** 0-3 Volume device type. 4-7 Maximum device block size. 8-9 Number of cylinders on this volume. 10-11 Number of tracks per cylinder on this volume. 12-13 Number of bytes per track on this volume. 14 Number of bytes required for gaps and check bits for each keyed block other than the last block on a track for this volume.[2] 15 Number of bytes required for gaps and check bits for the last keyed block on a track for this volume.[2] 16 Number of bytes to be subtracted for a block that is not keyed.[2] 17 Flags. Bits 0 through 6 are reserved. If bit 7 is set to 1, use tolerance factor on all blocks but the last to calculate the effective length of a block.[2] 18-19 Tolerance factor to be used in calculating the effective length of a block.
121 (79)	1	VOLRFLG	Volume record flags.
122 (7A)	1	SYSEXTDS	Number of extents per suballocation request allowed by the OS/VS system.
123 (7B)	4		Reserved.

The following field names identify information that is not contained in the volume catalog record; the information is derived from fields in the volume catalog record.

Offset	Bytes and Bit Pattern	Field Name	Description
	2	NODSPACE	Number of data spaces on the volume—a count of the Data Space Group sets of fields.
	2	NODSET	Number of data sets on the volume—a count of the Data Set Directory Entry sets of fields.

The following 6-byte entry contains control information for the sets of fields that follow it.

Offset	Bytes and Bit Pattern	Field Name	Description
127 (7F)	5		Pointer to the horizontal extension record. If this record is not continued on an extension record, this field contains zeros.
132 (84)	1		Number of set-of-fields pointers that follow. **Note:** The first set-of-fields pointer contains sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.

Volume Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
133 (85)	VL		6-byte pointers to sets of fields within the record. **Byte** **Meaning** 0 Reserved. 1-2 Displacement of the set of fields from the beginning of all sets of fields in this record. 3 Bits 0 and 1 are set to zero. If bit 1 is on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 of this byte and the sequence number in bytes 4 and 5, however, are kept. Bits 2 through 7 contain a type (group) code[4], describing the set of fields pointed to. 4-5 Sequence number of the set of fields pointed to.
	VL		6-byte pointers to sets offields contained in vertical Extension records. **Byte** **Meaning** 0-2 Control interval number of the extension record that contains this set of fields. 3 Bits 0 and 1 are set to B'10'. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to. 4-5 Sequence number of the set of fields pointed to.

[1] Zeros if the catalog is not recoverable or if there is no associated CRA volume.
[2] This value is used to calculate overhead bytes for keyed blocks to provide for compatibility with the DEVTYPE macro instruction. Blocks used by VSAM are, however, not keyed blocks.
[3] Fields describing (a) the volume's tracks' allocated/unallocated status, (b) each VSAM data space on the volume, and (c) each VSAM data set that resides in a VSAM data space are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in each set of fields.
[4] If the pointer is associated with track status (space map) information, the type (group) code is 5; with data space information, the type (group) code is 6; with a data set directory entry, the type (group) code is 8.

Space Map Set of Fields Format

The tracks on a VSAM volume are allocated to a VSAM object, or are unallocated, as described by the Space Map set of fields. Each bit position describes one track as allocated (bit = 0) or unallocated (bit = 1). This set of fields is associated with a pointer that contains a type (group) code of 5.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control Information: **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields (X'01'). 1 Hexadecimal displacement to the variable-length field, from the beginning of the set of fields (X'02').
2 (2)	VL	BITMAP	Portion of the volume bit map (1 to 440 bytes describing the allocated or unallocated status of 1 to 3520 direct-access device tracks).

Data Space Group Set of Fields Format

Each VSAM data space on the volume is described with a Data Space Group set of fields. This set of fields is associated with a pointer that contains a type (group) code of 6.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields (X'00'). 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields (X'55').
2 (2)	8	DSCBTS	Format-1 DSCB time stamp, which indicates when the DSCB was created. The time stamp is part of the name given to the Format-1 DSCB.
10 (A)	5	DSCBPTR	CCHHR of the Format-1 DSCB.

Volume Catalog Record Format: Data Space Group Set of Fields Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
15 (F)	1	SPHDFLG	Data-space flags.
	1...		Unique data space—this data space contains all or part of only one VSAM object.
	0...		Shared data space—this data space contains all or part of two or more VSAM objects.
	.1..		Automatically built data space—this data space was built as a result of an end of volume request for additional space.
	.... 1...		This data space was built when the user issued an Access Method Services DEFINE CATALOG command, and contains an OS/VS2 catalog.
	..xx .xxx		Reserved.
16 (10)	1	NODSPEXT	Number of extents in this data space.
17 (11)	1	DSPSOPT	Space options specified when the data space was created.
	10..		Track request—primary space allocation is specified in tracks.
	11..		Cylinder request—primary space allocation is specified in cylinders.
	..xx xxxx		Reserved.
20 (14)	3	DSPSSQ	Secondary space allocation quantity by which space is to be extended if required. This value is taken either from an Access Method Services DEFINESPACE command or from the first non-unique data set on this volume that caused space to be used.
23 (17)	64	SPEXTENT	Sixteen 4-byte extent descriptors in the form TTNN: TT starting track number of the extent (relative to the beginning of the volume). NN number of tracks in the extent.

Derived Data Space Information

The following field names identify information that is expected, but not contained in, the Data Space Group set of fields. The information is derived from fields in the volume catalog record.

Offset	Bytes and Bit Pattern	Field Name	Description
	1	SPHDFLG	Data space flags:
	..1.		A user catalog has extents within this data space—each CAXWA associated with the user contains the volume serial number of its catalog.
	...1		A master catalog has extents within this data space—the master catalog's volume is identified by a Data Set Directory Entry set of fields that contains a control interval value of 002.
	xx.. xxxx		Reserved.
	2	NODSDSP	Number of data sets in the data space—this information is derived by searching each Data and Index catalog record (pointed to by Data Set Directory Entry sets of fields and Cluster catalog records) for a volume information set of fields that contains the volume's serial number. Each set of fields so identified is searched to determine if the data set or index has been allocated space in one of the data space's extents.
The following field names refer to information about an extent of the data space:			
	2	TRKSUSED	Number of allocated tracks in the extent—the Space Map set of fields is scanned to determine the number of allocated tracks, based on the extent's starting track number and total number of tracks (contained in SPEXTENT).
	4	EXTSTART	Cylinder and track on which the extent begins—the extent's TT value (contained in SPEXTENT) is converted to a CCHH value.
	2	NOTRKEXT	Number of tracks in the extent—the extent's NN value (contained in SPEXTENT).
	2	SNSPHD	Sequence number of the set of fields that describes the extent's data space—the sequence number of the Data Space Group set of fields.
	VL	SPACEMAP	A variable-length space map that defines the allocated and unallocated space in the extent—the Space Map set of fields is converted to the format of this variable-length field based on the extent's starting track number and total length (contained in SPEXTENT).

Data Set Directory Entry Set of Fields Format

Each data set that resides in a VSAM data space on the volume is described with a Data Set Directory Entry set of fields. This set of fields is associated with a pointer that contains a type code of 8.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields (X'00'). 1 Hexadecimal displacement to the first variable-length field from the beginning of this set of fields (X'05').
2 (2)	3	DSIDNO	Control interval number of the Data or Index catalog record that describes this data set or index.
5 (5)	4	DSCRETS	Data/index identifier creation time stamp. Initialized by DEFINE and never altered. Consists of the high-order 4 bytes of the TOD clock value.
9 (9)	3	DSSUMTT	Sum of TT values converted from starting CCHHs of all extents of this data set on this volume.

Derived Data Set Information

The following field names identify information that is expected, but not contained in, the Data Set Directory Entry set of fields. The information is derived from fields in the volume catalog record.

Offset	Bytes and Bit Pattern	Field Name	Description
	1	MODSEXT	Number of data set extents on this volume.
	1	DSDIRFLG	Flags:
		Bit	**Meaning**
	1...		The Data or Index catalog record identifies the volume as a candidate volume—the Data Set's or Index's catalog record is searched; its volume information set of fields has zero extents.
	VL	DSSPSN	A variable-length collection of 3-byte fields that identify each data space within which the data set has extents allocated to it—this information is obtained by converting each volume information set of fields' extent descriptor's (EXTENT) SS value (data space extent's sequence number) so that the resulting 3-byte field is: **Byte** **Meaning** 0-1 Sequence number of the Data Space Group set of fields. 2 Number of extents (groups of contiguous tracks) assigned to the data set or index from the data space (limits: 1–255).

Extension Catalog Record Format

The Extension record contains overflow information from another catalog record.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record.
4 (4)	1	RELIND	Release indicator: 0=Nonenhanced VSAM; 1=Enhanced VSAM.
5 (5)	6	CRAVOL	CRA volume serial.[1]
11 (B)	3	CRAIDNO	CRA control interval number.[1]
14 (E)	4	CRADEVT	CRA device type.[1]
18 (12)	4	CRACRETS	CRA creation time stamp.
22 (16)	22		Zeros.
44 (2C)	1	ENTYPE	Record type—a "W" if this extension record is an extension of a Volume catalog record; an "E" if this extension record is an extension of any other type of catalog record.
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an extension record. Always zero for an extension catalog record.
48 (30)	1		Length of the fixed-length fields in the header fields, excluding any fixed length fields following displacement 49 (X'31'). This value is always equal to the displacement from the beginning of the record to the extension record's pointer.
The following 6-byte entry contains control information for the sets of fields that follow it.			
49 (31)	5		Pointer to the horizontal extension record. If this record is not continued on an extension record, this field contains zeros.
54 (36)	1		The number of set-of-fields pointers that follow.[2] **Note:** The first set-of-fields pointer contains sequence number = 0 and type code = 0. When bytes 0 to 2 are nonzero, this set-of-fields pointer points to the first (vertical) extension record that contains free space. The set-of-fields pointer is called the ASPT—available-space pointer. When bytes 0 to 2 are zero, the pointer is a dummy set-of-fields pointer—not available to be used to point to a set of fields.
55 (37)	VL		5-byte pointers to sets of fields within the record. If the extension record's identifier is "W" (the record extends a volume catalog record), these are 6-byte pointers. The sequence number in each 6-byte pointer is two bytes long.

Extension Catalog Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
			Byte **Meaning**
			0 Reserved.
			1-2 Displacement of the set of fields from the beginning of all sets of fields in this record.
			3 Bits 0 and 1 are set to zero. If bit 1 is on, the set of fields associated with this pointer has been deleted; the type (group) code in bits 2 through 7 of this byte and the sequence number in byte 4, however, are kept. Bits 2 through 7 contain a type (group) code describing the set of fields pointed to.[3]
			4 or 4-5 Sequence number of the set of fields pointed to, by type (group) code.[3]
	VL		5-byte pointers to sets of fields contained in a vertical Extension record. If the Extension record's identifier is "W" (the record extends a volume catalog record), these are 6-byte pointers. The sequence number in each 6-byte pointer is two bytes long.
			Byte **Meaning**
			0-2 Control interval number of the Extension record that contains this set of fields.
			3 Bits 0 and 1 are set to B'10'. Bits 2 through 7 contain a type (group) code that describes the set of fields pointed to.
			4 or 4-5 Sequence number of the set of fields pointed to, by type (group) code.[3]

1 Zeros if the catalog is not recoverable or if there is no associated CRA volume.

2 Fields describing (a) the volumes on which a data set resides, and (b) the password information associated with a data set are grouped into sets of fields. Pointers to each set of fields identify the type of information contained in each set of fields.

3 The sets of fields that are contained in an extension record depend upon the kind of catalog record that is extended. The format of the remainder of an extension record is, therefore, variable. The sets of fields in an extension record are, however, in the same format as they would be in the base record.

CRA Free Record Format

The CRA Free record indicates that the control interval in which it resides is free and points to the next control interval that is free because of deletion. Note that the Free CRA record is used only to mark a record that was used and has been deleted. The free space (control intervals) in the CRA that has never been assigned is not represented by Free CRA records. Control intervals 1,4,6,7, and 8 in the CRA are marked as Free records; however, their free control interval chain field is zero.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Reserved.
1 (1)	3	ENTIDNO	Control interval number of this record: X'00000N' where N=1,4,6,7, or 8.
4 (4)	40		Reserved.
44 (2C)	1	ENTYPE	Record type—"F."
45 (2D)	3		Free control interval chain field.
48 (30)	457		Reserved.

CRA Data Record Format

The CRA Data record describes the CRA data component of the CRA cluster. The CRA Data record, which is record type "D," occupies control interval 0 in the CRA.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record: X'000000'.
4 (4)	1	RELIND	Release indicator.
5 (5)	39		Zeros.
44 (2C)	1	ENTYPE	Record type—"D."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an extension record. Always zero.
48 (30)	1		Length of nonrepeating fixed-length fields. Always X'8F'.
49 (31)	44	ENTNAME	Name of the catalog that owns this CRA volume.
93 (5D)	8	OWNERID	Initialized to all X'FF'.
101 (65)	3	DSETCRDT	Date CRA was created. In packed-decimal form YDD.
104 (68)	3	DSETEXDT	Expiration date. Initialized to X'00000F'.
107 (6B)	1	ATTR1	Data set attributes: X'00'.
108 (6C)	1	ATTR2	Data set attributes: X'A0'.
109 (6D)	1	OPENIND	Open indicator: X'00'.
110 (6E)	4	BUFSIZE	Minimum buffer size: X'800'.
114 (72)	3	PRIMSPAC	Primary space. Initialized to number of tracks per cylinder.

CRA Data Record Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
117 (75)	3	SCONSPAC	Secondary space. Initialized to number of tracks per cylinder.
120 (78)	1	SPACOPTN	Space option: X'80'.
121 (79)	4	HURBADS	High-used RBA.
125 (7D)	4	HARBADS	High-allocated RBA.
129 (81)	4	LRECL	Logical record size: X'1F9'.
133 (85)	2	USERINFO	DOS user information: X'00'.
135 (87)	8	EXCPEXIT	Exception exit Initialized to all X'FF'.
The following entries contain control information for repeating fields:			
143 (8F)	5		Pointer to an extension record. Always zeros.
148 (94)	1		The number of set-of-field pointers that follow.
149 (95)	VL		Five-byte pointers to sets of fields.

AMDSB (Access Method Data Statistics Block) Set of Fields Format

The AMDSB set of fields contains a copy of the AMDSB control block that is updated each time the CRA is closed. This set of fields is associated with a pointer that contains a type (group) code of 1.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information. **Byte** **Meaning** 0 Count of the number of variable-length fields in this set of fields. 1 Displacement (X'62') to the first variable-length field from the beginning of this set of fields.
2 (2)	96	AMDSBCAT	Copy of the AMDSB. See "Data Areas" for detailed information about the AMDSB.

Association (Cluster) Set of Fields Format

There is one association in this entry, and it has a group occurrence sequence number of 1.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information: X'0006'.
2 (2)	1	TYPE	Record type—"C."
3 (3)	3	NAME	Control interval number: X'000002'.

Volume Information Set of Fields Format

There is one volume information set of fields in this entry. Its group occurrence sequence number is 1, and it resides in control interval number 5.

CRA Cluster Record Format

This record describes the CRA cluster. It is record type "C," and it occupies control interval 2 in the CRA.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record: X'000002'.
4 (4)	1	RELIND	Release indicator.
5 (5)	39		Zeros.
44 (2C)	1	ENTYPE	Record type—"C."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an extension record. Always zero.
48 (30)	1		Length of nonrepeating fixed-length fields: X'6C'.
49 (31)	44	ENTNAME	Name of the catalog that owns this CRA volume.
93 (5D)	8	OWNERID	Initialized to all X'FF'.
101 (65)	3	DSETCRDT	Date CRA was created. In packed-decimal form YDD.
104 (68)	3	DSETEXDT	Expiration date. Initialized to X'00000F'.
107 (6B)	1	CATTR	Cluster attributes: X'00'.
The following entries contain control information for repeating fields:			
108 (6C)	5		Pointer to an extension record. Always X'00'.
113 (71)	1		The number of set-of-field pointers that follow.
114 (72)	VL		Five-byte pointers to sets of fields.

Association (Data) Set of Fields Format

There is one association in this entry, and its group occurrence sequence number is 1.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information: X'0006'.
2 (2)	1	TYPE	Record type—"D."
3 (3)	3	NAME	Control interval number: X'000000'.

CRA Catalog Control Record Format

The catalog control record is used to manage CRA control interval allocation. It is record type "L," and it occupies control interval 3 in the CRA.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record: X'000003'.
4 (4)	1	RELIND	Release indicator.
5 (5)	39		Zeros.
44 (2C)	1	ENTYPE	Record type—"L."
45 (2D)	2		Number of the highest control interval within the current extents.
48 (30)	3		Number of the next free control interval that has not been previously assigned.
51 (33)	3		Number of deleted control intervals.
54 (36)	3		First deleted control interval in a chain of control intervals that are free because of deletion.
57 (39)	4		Reserved.
61 (3D)	4		CRA data high-used RBA.
65 (41)	4		CRA data high-allocated RBA.
69 (45)	436		Reserved.

CRA Data Extension Record Format

The Data Extension record is the extension of the CRA Data Record in control interval 0 of the CRA. The Data Extension record is record type "E," and it occupies control interval 5 in the CRA.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1		Zeros.
1 (1)	3	ENTIDNO	Control interval number of this record: X'000005'.
4 (4)	1	RELIND	Release indicator.
5 (5)	39		Zeros.
44 (2C)	1	ENTYPE	Record type—"E."
45 (2D)	2		Record length.
47 (2F)	1		Number of variable-length fields that precede the pointer to an extension record. Always zero.
48 (30)	1		Length of nonrepeating fixed-length fields: X'6C'.
The following entries contain control information for repeating fields:			
49 (31)	5		Pointer to an extension record. Always zeros.
54 (36)	1		The number of sets-of-fields pointers that follows.
55 (37)	VL		Five-byte pointers to sets of fields within the record.

Volume Information Set of Fields Format

There is one volume information set of fields in this entry, and its group occurrence sequence number is 1.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2		Control information: X'0327'.
2 (2)	4	DEVTYP	Device type.
6 (6)	6	VOLSER	Volume serial number.
12 (C)	2	FILESEQ	File sequence number.
14 (E)	1	VOLFLG	Volume flags: X'80'.
15 (F)	1	NOEXTNT	Number of extents.
16 (10)	4	HKRBA	High-key RBA.
20 (14)	4	HURBA	High-used RBA.
24 (18)	4	HARBA	High-allocated RBA.
28 (1C)	4	PHYBLKSZ	Block size: X'200'.
32 (20)	2	NOBLKTRK	Number of blocks per track.
34 (22)	2	NOTRKAU	Number of tracks per allocation unit. Initialized to number of tracks per cylinder.
36 (24)	1	ITYPEXT	Type-of-extent indicator.
37 (25)	2	DSDIRSN	Data set directory sequence number.
39 (27)	2	LOKEYV	Low key length on volume.
41 (29)	2	HIKEYV	High key length on volume.
43 (2B)	2	EXTENT	Length of extent information.
45 (2D)	VL		20-byte extent descriptors.

Field Name Dictionary

The field name dictionary is an internal data area that provides a map between field names and fields within catalog records, as well as information that is not within catalog records. The dictionary also allows the dictionary user to specify values (for example, the number of sets of fields to be processed) by associating them with a dictionary name. A field name is specified in a CTGFL (field parameter list). For a description of the CTGFL, see "Data Areas." The catalog management modules reference the field name dictionary for the location, length, and type of fields.

The field name dictionary is a series of eight-byte entries. In addition, there is an index of combination field names. Each combination field name allows catalog management to locate more than one field at a time.

The field-name dictionary is located in module IGG0CLAY.

Field Name Dictionary Entry Format (continued)

Offset	Bytes and Bit Pattern	Description
0	4	Shortened field name: the first, second, fifth, and sixth characters of the eight-character field name.
4	1	Flags that describe the field.
	000.	The field is fixed-length and appears in the header portion of a record (before the Extension record pointer).
	001.	A combination field name.[1]
	010.	The field is fixed-length and is part of a set of fields that follows the Extension record pointer.
	100.	The field is variable-length and appears in the header portion of a record (before the Extension record pointer).
	110.	The field is variable-length and is part of a set of fields that follows the Extension record pointer.
	011.	Special field.[2]
	111.	Special field.[2]
	...0	Not a flag field, which means that a CLC (compare logical character) instruction can be used to test this field.
	...1	Flag field, which means that a TM (test under mask) instruction can be used to test this field.
	.... 1...	A fixed-length field within a variable-length field in a set of fields.
	.... 0...	Not a fixed-length field within a variable-length field in a set of fields.
	.... .1..	CRA updates are to be suppressed.
	.... .0..	CRA updates are not suppressed.
	.... ..1.	This field must be retrieved from the upgrade set.
	.... ..0.	This field has no special retrieval characteristics.
	x	Reserved.
5	1	Bytes that identify the location of the field:

Type of field name	Contents of this byte
Fixed-length:	Displacement in bytes from
in the header:	the beginning of the record.
in a set of fields:	the beginning of the set of fields.
in a group of fixed-length fields within a variable-length field:	Length of the group of fixed-length fields.
Variable-length:	Zero.
Combination:	Index value in the combination-name index.

Offset	Bytes and Bit Pattern	Description
6	1	Bytes that identify the location of the field (continued):

Type of field name	Contents of this byte
Fixed-length:	Length of the field (in bytes).
Variable-length:	Sequence number of the field.
Combination:	Number of fields identified by the combination name.

Field Name Dictionary Entry Format (continued)

Offset	Bytes and Bit Pattern	Description
7	1	A code that indicates which group of data (the kind of catalog record and the set of fields) this field is in.

Set of Fields	Type Codes
Type code:	Description:
0	Header field
1	AMDSB
2	Association
3	Volume information
4	Password
5	Space map [3]
6	Data space group [3]
7	Reserved
8	Data set directory[3]

[1] "Combination field name" indicates that the name supplied is a name that allows catalog management to locate a group of related fields.
[2] The field-name dictionary permits catalog management to locate information that is not contained in catalog records (for example, derived information).
[3] This set of fields is contained only in a Volume catalog record.

Bytes 4 through 7 of the field-name dictionary record describe the field. When a caller makes a request in a CTGFL, dictionary information is moved from the dictionary to the CTGFL.

To clarify the use of the dictionary as a means of gaining access to catalog information, refer to the examples that follow.

Combination Field Names

A combination field name identifies a group of related fields. When a catalog management user requires information from many catalog record fields (eg. all fields in a set of fields), the user builds a CTGFL that contains a combination field name. The combination field name in the CTGFL identifies an entry in the field name dictionary (in module IGG0CLAY). The entry identifies the field name as a combination field name, specifies the number of fields contained in the combination, and points to the starting point for the combination in the combination field name index. The combination field name index contains a group of 1-byte entries. Each entry points to an entry in the field name dictionary, as shown in Figure 41, Resolution of a Combination Field Name. The entry in the field name dictionary describes one of the fields identified by the combination.

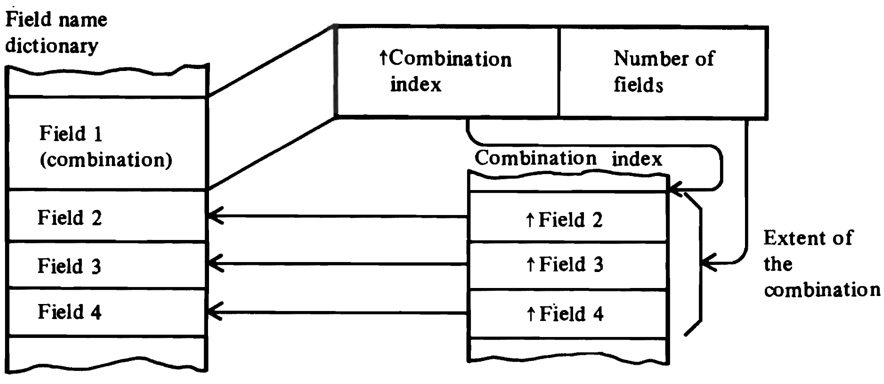


Figure 41. Resolution of a Combination Field Name

Field Name Dictionary Entries

This section lists the field name dictionary entries in alphabetic order. IGG0CLAY lists the field name dictionary entries in alphabetic order, too, but that list is ordered by *abbreviated* field name (first, second, fifth, and sixth characters of the field name). The field's name, length, location, and description are listed here.

The field's location is coded as follows:

Code	Location description
AMDSB	The AMDSB set of fields
Assoc	The association set of fields
Combin	A combination field name
Header	The catalog record's header fields. Note: some header fields appear in all catalog records; other header fields appear only in one (or more) type of catalog record.
Password	The password set of fields
Special	A piece of information that is derived from information in catalog record fields or catalog control blocks, but is not stored in a catalog record
Vol Info	The volume information set of fields
Vol–Dir	The data set directory entry set of fields in the volume catalog record
Vol–DSG	The data space group set of fields in the volume catalog record
Vol–SM	The space map set of fields in the volume catalog record

Notes

- The list of field names following each combination name in this section is ordered alpha-numerically. See module listing IGG0CLAY for the actual order of the combination's field names.
- Each variable-length field contains two bytes of control information followed by a number of data bytes. The 2-byte control information specifies the total length of the field.
- The field's length (len) is:

  'n' a number of bytes.

  'VL' indeterminate, because the field is a variable length field.

  '—' indeterminate, because the combination name group of fields includes one or more variable-length fields.

Field Name	Len	Location	Description
AMDCIREC	8	AMDSB	AMDSB control-interval size and maximum logical record size (show catalog support); includes fields AMDCINV and AMDLRECL
AMDKEY	4	AMDSB	AMDSB relative key position and key length (show catalog support); includes fields AMDRKP and AMDKEYLN
AMDSBCAT	96	AMDSB	Copy of the AMDSB control block
AMDSBSC	12	Combin	AMDSB fields for show catalog Includes fields: AMDCIREC, AMDKEY
AMDSB1	6	AMDSB	Part of AMDSB
AMDSB2	10	AMDSB	Part of AMDSB
AMDSB3	68	AMDSB	Part of AMDSB

Field Name Dictionary Entries (continued)

Field Name	Len	Location	Description
ASSOCSC	8	Combin	Twin associations for show catalog support
ATTR1	1	Header	Data set attributes
ATTR2	1	Header	Data set attributes
BITMAP	VL	Vol–SM	Volume space map showing the allocated and unallocated tracks on a direct-access volume
BUFSIZE	4	Header	Minimum buffer size
CATACB	4	Special	Address of the catalog's ACB control block
CATTR	1	Header	Cluster attributes
CATVOL	15	Combin	Volume information set of fields for a nonVSAM data set's catalog record Includes fields: DEVTYP, FILESEQ, RELREPNO, VOLFLG, VOLSER
CNTREPNO	2	Special	Maximum number of RELREPNOs to be processed
CRACRETS	4	Header	CRA creation time stamp
CRADEVT	4	Header	CRA device type
CRADIRCT	12	Combin	Data set directory fields for catalog recovery Includes fields: DSCRETS, DSIDNO, DSSUMTT, RELREPNO
CRADITS	4	Header	Data/index identifier creation time stamp
CRAIDNO	3	Header	CRA control interval number
CRAVOL	6	Header	CRA volume serial
DATASPAC	85	Combin	Data space group set of fields Includes fields: DSCBPTR, DSCBTS, DSPSOPT, DSPSSQ, NODSPEXT, SPEXTENT, SPHDFLG, RELREPNO
DEVTYP	4	Vol Info	Device type
DEXTENTS	68	Combin	All fields in the data space group set of fields required by Suballocate processing Includes fields: NODSPEXT, SPEXTENT, SPHDFLG, RELREPNO
DIRECTRY	5	Combin	Data set directory entry set of fields Includes fields: DSIDNO, RELREPNO
DSATRO	3	Combin	All data set attributes flags fields Includes fields: ATTR1, ATTR2, OPENIND
DSATTR	2	Combin	All data set attributes flags fields, except the open indicator Includes fields: ATTR1, ATTR2, OPENIND
DSCBPTR	5	Vol–DSG	TTR of the DSCB that describes the data space in the volume's VTOC
DSCBTS	8	Vol–DSG	Data space timestamp
DSCBTTR	3	Vol Info	TTR of the format 1 (identifier) DSCB that describes the space allocated to a nonVSAM data set, if the entire data set resides on one volume.
DSCRETS	4	Vol–Dir	Data/index identifier time stamp

Field Name Dictionary Entries (continued)

Field Name	Len	Location	Description
DSDIRECT	—	Combin	Data set directory entry set of fields, including some of the related derived fields
			Length: 7 + L'DSSPSN
			Includes fields: DSDIRFLG, DSIDNO, DSSPSN, NODSEXT, RELREPNO
DSDIRFLG	1	Special	Data set directory flags (derived)
DSDIRSN	2	Vol Info	Data set directory sequence number in the volume information set of field's extent descriptor
DSETCRDT	3	Header	Data set creation date
DSETEXDT	3	Header	Data set expiration date
DSIDNO	3	Vol–Dir	Control interval number of the data set directory entry's object's catalog record
DSPDSCRP	13	Combin	Space descriptor set of fields (a group of derived fields)
			Includes fields: EXTSTART, NOTRKEXT, RELREPNO, SNSPHD, SPACEMAP, TRKSUSED
DSPSOPT	1	Vol–DSG	Space options for a data space
DSPSSQ	3	Vol–DSG	Secondary data space quantity
DSSPSN	VL	Special	Data space sequence numbers for the data set directory entry set of fields (derived)
DSTYPNAM	4	Combin	Catalog record type and dsname
			Includes fields: ENTIDNO, ENTYPE
DSSUMTT	3	Vol–Dir	Sum of starting tracks of all extents of data set on this volume
ENTASSOC	8	Combin	Association set of fields
			Includes fields: NAME, RELREPNO, TYPE
ENTIDNO	3	Header	Control interval number of the catalog record
ENTNAME	44	Header	Dsname in the catalog record
ENTUPGD	12	Combin	Combination for associations in the upgrade set
			Includes fields: NAME, NAME2, RELREPNO, TYPE, TYPE2
ENTVOL	37	Combin	Volume information set of fields
			Includes fields: DEVTYP, FILESEQ, HARBA, HKRBA, HURBA, ITYPEXT, NOBLKTRK, NOEXTNT, NOTRKAU, PHYBLKSZ, VOLFLG, VOLSER, RELREPNO, DSDIRSN, EXTENT, HIKEYV, LOKEYV
ENTYPE	1	Header	Catalog record type identifier
EXCPEXIT	8	Header	Exception exit
EXTENT	VL	Vol Info	Extent descriptors
EXTSTART	4	Special	Starting point in the list of data space extents in the data space group set of fields (derived)

Field Name Dictionary Entries (continued)

Field Name	Len	Location	Description
EXTVOL	—	Combin	Volume information set of field's fields required for VSAM End of Volume processing Length: 7 + L'EXTENT Includes fields: DEVTYP, EXTENT, ITYPEXT, RELREPNO
FILESEQ	2	Vol Info	File sequence number for a nonVSAM data set
GDGATTR	1	Header	GDG attributes flag
GDGLIMIT	1	Header	Maximum number of GDG levels allowed
GDGLVLIM	—	Combin	Fields required to determine the number of cataloged generations in a GDG (generation data group) Length: 1 + L'GDGLVLS Includes fields: GDGLIMIT, GDGLVLS
GDGNAME	8	Combin	Association set of fields for a generation data group (GDG) Includes fields: GENLEVEL, NAME, TYPE
GENDSP	44	Special	Generated data space dsname
GENLEVEL	4	Assoc	GDG generation level—GGGG
GENLVLS	VL	Header	Generation level difference string
HARBA	4	Vol Info	High-allocated RBA on the volume for the data set
HARBADS	4	Header	High-allocated RBA for the data set (not always the same as HARBA if the data set resides on several volumes)
HIKEYV	2	Vol Info	Key-sequenced data set's high key value on a volume or, if the data set is divided into key ranges, the key range's high key value
HKRBA	4	Vol Info	RBA of the record containing the high key of a key-sequenced data set on a volume or, if the data set is divided into key ranges, the key range's high key value
HKURBA	8	Combin	Data set's high-key and high-used RBAs Includes fields: HKRBA, HURBA
HURBA	4	Vol Info	High-used RBA on the volume for the data set
HURBADS	4	Header	High-used RBA for the data set (not always the same as HURBA if the data set resides on several volumes)
ITYPEXT	1	Vol Info	Type of extent indicator
LOKEYV	VL	Vol Info	Key-sequenced data set's low-key value on a volume or, if the data set is divided into key ranges, in the key range
LRECL	4	Header	Average logical record size
MAPSPACE	—	Combin	Volume catalog record's space map set of fields Length: 2 + L'BITMAP Includes fields: BITMAP, RELREPNO
NAME	3	Assoc	Control interval number
NAME2	3	Assoc	Control interval number of index in twin association of upgrade set

Field Name Dictionary Entries (continued)

Field Name	Len	Location	Description
NAMEDS	4	Combin	Association set of fields
			Includes fields: NAME, TYPE
NOBLKTRK	2	Vol Info	Number of blocks per track
NOBYTAU	4	Vol Info	Number of bytes per allocation unit
NOBYTTRK	4	Vol Info	Number of bytes per track
NODSDSP	2	Special	Number of data sets in a data space (derived)
NODSET	2	Special	Number of data sets on a volume (derived)
NODSEXT	1	Special	Number of data set directory extents (derived)
NODSPACE	2	Header	Number of data spaces on a volume (derived)
NODSPEXT	1	Vol–DSG	Number of data space extents
NOEXTNT	1	Vol Info	Number of volume information set of field's extents
NONVOL	16	Combin	NonVSAM data set's volume information set of fields
			Includes fields: DEVTYP, DSCBTTR, FILESEQ, RELREPNO, VOLFLG, VOLSER
NOTRKAU	2	Vol Info	Number of tracks per allocation unit
NOTRKEXT	2	Special	Number of tracks in a data space's extent (derived)
OPENCNT	1	Header	Open count
OPENC2	61	Combin	Catalog fields required by OPEN
			Includes fields: BUFSIZE, ENTNAME, EXCPEXIT, HURBADS, SPACOPTN
OPENIND	1	Header	Open indicator
OPNCALL1	53	Combin	All header fields required by VSAM Open processing
			Includes fields: BUFSIZE, ENTNAME, HURBADS, SPACOPTN
OWNERID	8	Header	Owner identification number
PASSATMP	2	Password	Number of attempts the operator has to supply the correct password
PASSPRMT	8	Password	Data set's prompting name (codename) for security verification
PASSWALL	—	Combin	Password set of fields
			Length: 50 + L'USERAREC
			Includes fields: PASSATMP, PASSPRMT, PASSWORD, USERAREC, USVRMDUL
PASSWORD	32	Password	All (4) 8-byte passwords
PHYBLKSZ	4	Vol Info	Physical blocksize
PRIMSPAC	3	Header	Primary space allocation amount
RELCRA	14	Combin	Release indicator and CRA header fields
			Includes fields: CRADEVT, CRAIDNO, CRAVOL, RELIND
RELIND	1	Header	Release indicator
RELREPNO	2	Special	Relative repetition number
REPNO	2	Special	Highest nondeleted sequence number
RGATTR	1	Header	Path/alternate index indicator

Field Name Dictionary Entries (continued)

Field Name	Len	Location	Description
SCONSPAC	3	Header	Secondary space allocation amount requirement
SECFLAGS	1	Header	RACF security attributes
SLOCVOL	19	Combin	Fields required by Superlocate Includes fields: DEVTYP, DSCBTTR, FILESEQ, ITYPEXT, RELREPNO, VOLFLG, VOLSER
SNSPHD	2	Special	Data space sequence number for a data space (derived)
SPACEHDR	23	Combin	Data space group set of fields Includes fields: DSCBPTR, DSCBTS, DSPSOPT, DSPSSQ, NODSDSP, NODSPEXT, RELREPNO, SPHDFLG
SPACEMAP	1	Special	Data space run length codes (derived)
SPACOPTN	1	Header	Space options
SPACPARM	7	Combin	Space allocation quantities and options Includes fields: PRIMSPAC, SPACOPTN, SCONSPAC
SPEXTENT	64	Vol–DSG	Data space extent descriptors
SPHDFLG	1	Vol–DSG	Data space flag (partially derived)
SYSEXTDS	1	Header	Number of extents allowed per suballocation in the volume catalog record
TOENTVOL	43	Combin	Track overflow entire volume Includes fields: DEVTYP, DSDIRSN, FILESEQ, HARBA, HKRBA, HURBA, ITYPEXT, NOBLKTRK, NOBYTTRK, NOEXTNT, NOTRKAU, PHYBLKSZ, RELREPNO, VOLFLG, VOLSER, LOKEYV, HIKEYV, EXTENT, NOBYTAU
TOVOLDEV	20	Combin	Track overflow volume and device characteristics Includes fields: DEVTYP, NOBLKTRK, NOBYTAU, NOBYTTRK, NOTRKAU, PHYBLKSZ
TRBAEXT	3	Vol Info	Test RBA for EOV mount by RBA
TRKSUSED	2	Special	Tracks used in the data space (derived)
TYPE	1	Assoc	Association entry type
TYPE2	1	Assoc	Entry type for second name of a twin association in the upgrade set
UPDVOL	—	Combin	Volume information set of fields for UPDATE–Extend processing Length: 43 + L'HIKEYV + L'LOKEYV Includes fields: DEVTYP, DSDIRSN, FILESEQ, HARBA, HIKEYV, HKRBA, HURBA, ITYPEXT, LOKEYV, NOBLKTRK, NOEXTNT, NOTRKAU, PHYBLKSZ, RELREPNO, VOLFLG, VOLSER
UPGRADE	8	Combin	Fields for a twin association in the upgrade set Includes fields: TYPE, NAME, TYPE2, NAME2
USERAREC	VL	Password	Additional security verification data
USVRMDUL	8	Password	USVR (user security verification routine) module name
VOLDEV	12	Combin	Volume information set of fields required by VSAM Open processing Includes fields: DEVTYP, NOBLKTRK, NOTRKAU, PHYBLKSZ
VOLDVCHR	20	Header	Device characteristics—in the volume catalog record
VOLEXT	—	Combin	Volume information set of fields Length: 10 + L'EXTENT Includes fields: DSDIRSN, EXTENT, RELREPNO, VOLSER
VOLFLG	1	Vol Info	Volume information flags

Field Name Dictionary Entries (continued)

Field Name	Len	Location	Description
VOLPHY	—	Combin	Volume information set of fields required by VSAM Open processing Length: 23 + L'EXTENT + L'HIKEYV + L'LOKEYV Includes fields: EXTENT, HARBA, HIKEYV, HKRBA, HURBA, ITYPEXT, LOKEYV, NOEXTNT, RELREPNO, VOLFLG, VOLSER
VOLRFLG	1	Header	Volume catalog record flags
VOLSER	6	Vol Info	Volume serial number
VOLTSTMP	8	Header	Volume catalog record timestamp

Dictionary Example 1

The DSETCRDT (data set creation date) field is written as a parameter string of the macro, as follows:

DSETCRDT,0,101,3,0

It appears in the dictionary as X'C4E2C3D900650300'. The first X'00' (B'0000 0000') is the fourth byte value of the field-name dictionary entry; it indicates that DSETCRDT is (a) a fixed-length field, (b) not part of a set of fields, and (c) not a flag field.

The X'65' is the fifth-byte value of the field-name dictionary entry; it indicates that DSETCRDT is at displacement X'65' from the beginning of the record in which it appears.

The 3 is the sixth-byte value of the entry; it indicates that DSETCRDT is three bytes long.

The 0 is the seventh-byte value of the entry; it is zero because DSETCRDT is not part of a set of fields and, therefore, is not associated with a set of fields type (group) code.

Dictionary Example 2

The DSPSOPT (data-space-creation space options) field is written as a parameter string of the macro as follows:

DSPSOPT,80,19,1,6

It appears in the dictionary as X'C4E2D6D750110106'. The X'50' (B'0101 0000') is the fourth-byte value of the field-name dictionary entry; it indicates, when converted to binary, that DSPSOPT is (a) a fixed-length field that is part of a set of fields, (b) a flag field, and (c) not a fixed-length field within a variable-length field.

The X'13' is the fifth-byte value of the field-name dictionary entry; it indicates that DSPSOPT is at displacement X'13' from the beginning of the set of fields to which it belongs.

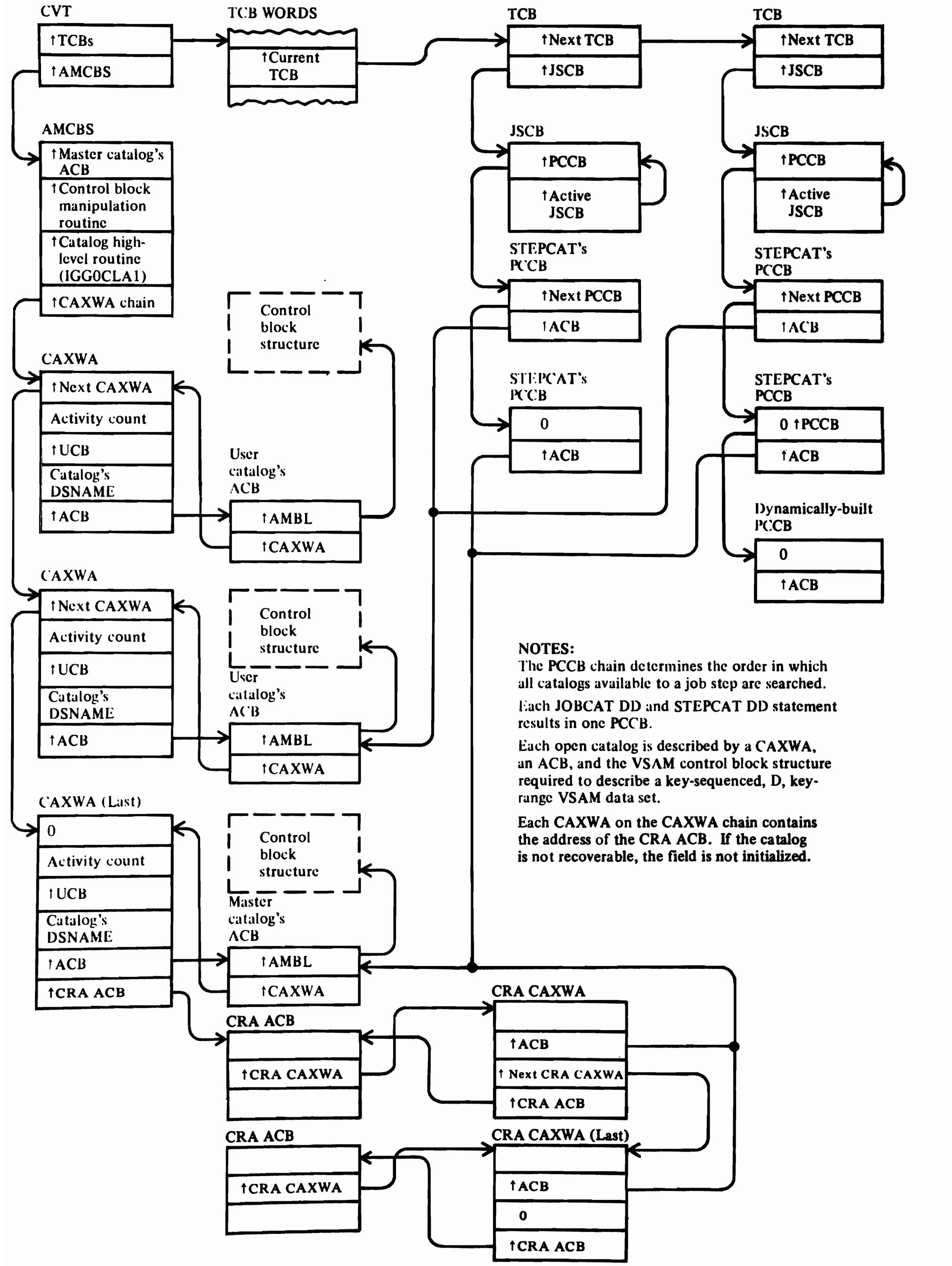
The 1 is the sixth-byte value of the entry; it indicates that DSPSOPT is 1 byte long.

The 6 is the seventh-byte value of the entry; it indicates that DSPSOPT is part of a set of fields associated with a type (group) code of 6, which means that it is part of a set of fields that contains VSAM data space information.

Control Block Interrelationships

Figure 42, Open Catalog Control Blocks, shows the OS/VS2 system and catalog management control blocks that describe an OS/VS2 catalog to the OS/VS2 system.

Figure 43, VSAM Control Blocks That Describe a Catalog (A Key-Sequenced Key-Range VSAM Data Set), shows the VSAM control blocks that describe the OS/VS2 catalog as a VSAM data set. This control block structure allows VSAM Record Management to read and write control intervals in the OS/VS2 catalog, and to update the catalog's index, as required when OS/VS2 catalog management I/O functions issue GET, PUT, and ERASE macro instructions. See *OS/VS2 VSAM Logic* for VSAM Record Management details.



NOTES:

The PCCB chain determines the order in which all catalogs available to a job step are searched.

Each JOBCAT DD and STEPCAT DD statement results in one PCCB.

Each open catalog is described by a CAXWA, an ACB, and the VSAM control block structure required to describe a key-sequenced, D, key-range VSAM data set.

Each CAXWA on the CAXWA chain contains the address of the CRA ACB. If the catalog is not recoverable, the field is not initialized.

Figure 42. Open Catalog Control Blocks

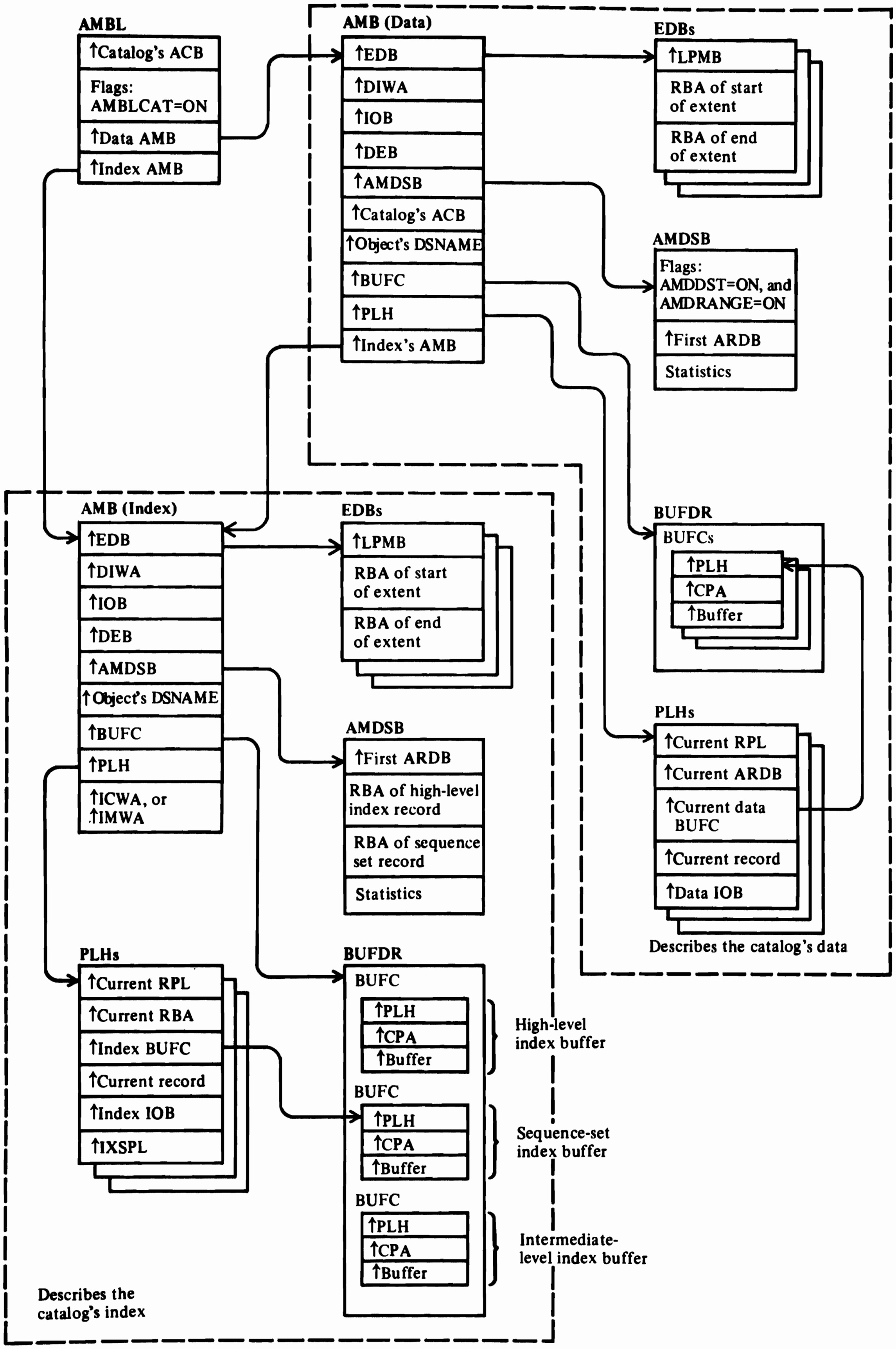


Figure 43. VSAM Control Blocks That Describe a Catalog (Key-Sequenced Key-Range VSAM Data Set)

Figure 44 shows the OS/VS2 catalog management control blocks built when OS/VS2 catalog management is called to process a catalog record.

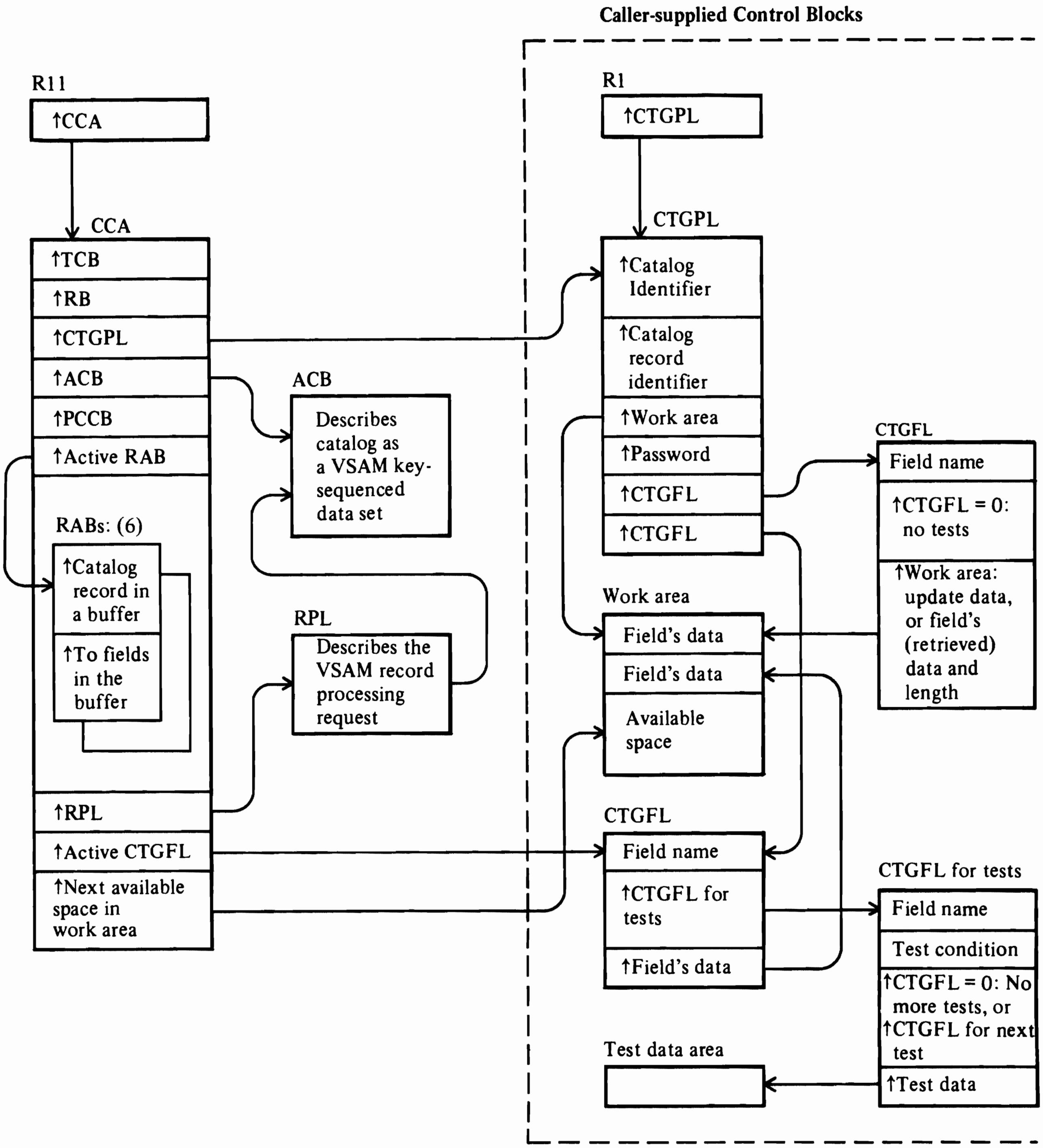


Figure 44. Catalog Management Control Blocks

Catalog Management Control Block Descriptions

ACB—Access Method Control Block

The VSAM ACB describes a VSAM cluster or OS/VS2 catalog. After the OS/VS2 catalog is opened, its ACB is pointed to by a CAXWA (CAXACB). The master catalog's ACB is pointed to by the AMCBS (CBSACB) and by the last CAXWA in the CAXWA chain.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1	ACBID	Control block identifier, X'A0'
1 (1)	1	ACBSTYP	Subtype
2 (2)	2	ACBLENG	Length of the ACB
4 (4)	4	ACBAMBL	Address of the AMBL
		ACBJWA	Address of the JES work area
		ACBIBCT	Address of the IBCT
8 (8)	4	ACBINRTN	Address of the VSAM Interface routine (IDA019R1)
12 (C)	2	ACBMACRF	MACRF flags
	Byte 1	ACBMACR1	MACRF flag byte 1:
	1...	ACBKEY	The record is identified by a key—keyed processing
	.1..	ACBADR	The record is identified by a RBA
		ACBADD	(relative byte address)—addressed processing
			..1.....ACBCNVControl interval processing
		ACBBLK	
	...1	ACBSEQ	Sequential processing
	.... 1...	ACBDIR	Direct processing
	.... .1..	ACBIN	Input (GET, READ) processing
	.... ..1.	ACBOUT	Output (PUT, WRITE) processing
	1	ACBUBF	User-supplied buffer space
13 (D)	Byte 1	ACBMACR2	MACRF flag byte 2:
	xxx.		Reserved
	...1	ACBSKP	Skip sequential processing
	.... 1...	ACBLOGON	VTAM LOGON indicator
	.... .1..	ACBRST	Set data set to empty state
	.... ..1.	ACBDSN	Basic subtask shared control block connection on common DSNAMEs
	1	ACBAIX	Entity to be processed is the alternate index of the path specified in the given DDNAME
14 (E)	1	ACBBSTNO	Number of concurrent strings for alternate index path
15 (F)	1	ACBSTRNO	Number of RPL strings
16 (10)	2	ACBBUFND	Number of buffers requested for data
18 (12)	2	ACBBUFNI	Number of buffers requested for index

Access Method Control Block (ACB) Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
20 (14)	4	ACBBUFPL	Address of the buffer header (BUFC)
	1	ACBMACR3	MACRF flag byte 3:
	x...		Reserved
	.1..	ACBLSR	Local shared resources
	..1.	ACBGSR	Global shared resources
	...1	ACBICI	Improved control interval processing
	.... 1...	ACBDFR	Write operations are to be deferred
	.... .1..	ACBSIS	Sequential insert strategy
	.... ..0.	ACBNCFX	Control blocks are not fixed
	.... ..1.		Control blocks are fixed
	x		Reserved
	1	ACBMACR4	Reserved
22 (16)	2	ACBJBUF	Number of buffers requested for journal
24 (18)	1	ACBRECFM	Record format:
	1...	ACBRECAF	JES format
	.xxx xxxx		Reserved
25 (19)	1	ACBCCTYP	Control character:
	xxxx		Reserved
	 xxxx	ACBASA	Control character type
26 (1A)	2	ACBOPT ACBDSORG ACBCROPS	Non-user options Match ACBDORGA with DCBDSORG Checkpoint/restart options:
	1...	ACBCRNCK	Restart has not checked for modifications since last checkpoint
	.1..	ACBCRNRE	Data added since last checkpoint has not been erased by restart, and no reposition to last checkpoint takes place
	.... 1...	ACBDORGA	ACB indicator
	.... ..1.	ACBSDS	The ACB describes a data set that can be accessed by more than one address space's programs.
	..xx .x.x		Reserved
28 (1C)	4	ACBMSGAR	Message area
32 (20)	4	ACBPASSW	Address of the user-supplied password
36 (24)	4	ACBEXLST ACBUEL	Address of the user exit list (EXLST) Alternate name for ACBEXLST
Before OPEN			
40 (28)	8	ACBDDNM	DD name
After OPEN			
40 (28)	2	ACBTIOT	Offset to the TIOT
42 (2A)	1	ACBINFL	Indicator flags
43 (2B)	1	ACBAMETH	Access method type
44 (2C)	1	ACBERFL	Error flags
45 (2D)	3	ACBDEB	Address of the DEB

Access Method Control Block (ACB) Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
Not Changed by OPEN			
48 (30)	1	ACBOFLGS	Open/Close flags:
	..1.	ACBEOV	EOV concatenation
	...1	ACBOPEN	The ACB is open
	.... 1...	ACBDSERR	No further requests are possible against the ACB
	.... ..1.	ACBEXFG	An ACB Exit routine exists
	1	ACBIOSFG	The Open or Close routine is in control
	xx.. .x..		Reserved
49 (31)	1	ACBERFLG	Error flags **Note:** See "Diagnostic Aids: Error Codes" in *OS/VS2 VSAM Logic* for details on the ACBERFLG error flags.
50 (32)	2	ACBINFLG	Indicator flags:
	.1..	ACBJEPS	JEPS processing
	..1.	ACBIJRQE	RQE being held by JAM
	...1	ACBCAT	The ACB describes a VSAM catalog
	.... 1...	ACBSCRA	Catalog control blocks are built in system storage
	.... .1..	ACBUCRA	Catalog control blocks are built in user storage
	.... ..1.	ACBVVIC	Data set being opened can be accessed by more than one address space's programs
	1	ACBBYPSS	Bypass security on OPEN if user authorized.
	x...		Reserved
51 (33)	1		Reserved
52 (34)	4	ACBUJFCB	Address of the user JFCB
56 (38)	4	ACBBUFSP	Amount of space available for the buffers
60 (3C)	2	ACBBLKSZ ACBMSGLN	Length of the physical DASD record Message area length
62 (3E)	2	ACBLRECL	Length of the user's record
64 (40)	4	ACBUAPTR	Address of the user's work area
68 (44)	4	ACBCBMWA	Address of the CBM work area
72 (48)	4	ACBAPID	Address of application ID

AMCBS—Access Method Control Block Structure Block

The AMCBS contains information that is used by OS/VS to locate the master catalog and user catalogs. The AMCBS is built when the master catalog is opened, during NIP (nucleus initialization processing). The CVT (CVTCBSP) points to the AMCBS.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2	CBSID	AMCBS ID character
2 (2)	2	CBSSIZ	Length of the AMCBS
4 (4)	4	CBSMCSTA	Location (CCHH) of the master catalog
8 (8)	4	CBSACB	Address of the OS/VS2 master catalog's ACB
12 (C)	4	CBSCBP	Address of the control block manipulation routine (IDA019C1)
16 (10)	4	CBSCMP	Address of the catalog management high-level routine (IGG0CLA1)
		CBSMCUCB	Address of the master catalog's UCB (contains this information until the master catalog is opened)
20 (14)	4	CBSCAXCN	Address of the CAXWA chain
24 (18)	4	CBSCRACA	Address of the CRA CAXWA chain
28 (1C)	4	CBSCRTCB	Address of CRA task TCB
32 (20)	64	CBSVSRT	CDS (compare and swap double) word for VSRT (VSAM shared resource table)
The following pair of fields is repeated eight times—once for each of the keys 0 through 7:			
	4	CBSVUSE	VSRT use count
	4	CBSVPTR	Address of the VSRT

AMDSB—Access Method Data Statistics Block

The AMDSB contains statistical information about record processing in the data set. It also contains some of the data set's attributes and specifications. The AMDSB is built, using the data set or index catalog record's AMDSB set of fields, when the cluster is opened. A copy of the AMDSB is contained in the data set's Data catalog record and, if the data set is key-sequenced, in the Index catalog record. The data or index AMB (AMBDSB) points to its associated AMDSB.

Access Method Data Statistics Block (AMDSB)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1	AMDSBID	AMDSB identifier, X'60'
1 (1)	1	AMDATTR	Attributes of the data set:
	1...	AMDDST	Key-sequenced data set
	0...		Entry-sequenced data set
	.1..	AMDWCK	Check each record when it is written
	..1.	AMDSDT	Sequence set is stored with the data and is replicated—duplicated as many times as possible around the track
	...1	AMDREPL	All index records are replicated—duplicated as many times as possible around the track
	.... 1...	AMDORDER	Use the volumes in the same order as in the volume list
	.... .1..	AMDRANGE	The data set is divided into key ranges
	.... ..1.	AMDRRDS	Relative record data set
	1	AMDSPAN	Data set contains spanned records
2 (2)	2	AMDLEN	Length of the AMDSB
4 (4)	2	AMDNEST	Number of index entries in the index section
		AMDAKRKP	Relative key position of the alternate key.
6 (6)	2	AMDRKP	Relative key position
8 (8)	2	AMDKEYLN	Key length
10 (A)	1	AMDPCTCA	Percentage of free control intervals in the control area
11 (B)	1	AMDPCTCI	Percentage of free bytes in the control interval
12 (C)	2	AMDCIPCA	Number of control intervals in a control area
14 (E)	2	AMDFSCA	Number of free control intervals in a control area
16 (10)	4	AMDFSCI	Number of free bytes in a control interval
20 (14)	4	AMDCINV	Control interval size
24 (18)	4	AMDLRECL	Maximum record size
28 (1C)	4	AMDHLRBA	Relative byte address (RBA) of the high-level index record (KSDS)
		AMDNSLOT	Number of record slots per control interval (RRDS)
32 (20)	4	AMDSSRBA	Relative byte address (RBA) of the first sequence-set record (KSDS)
		AMDMAXRR	Maximum valid relative record number (RRDS)
36 (24)	4	AMDPARDB	Address of the first ARDB

Access Method Data Statistics Block (AMDSB)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
40 (28)	1	AMDATTR3	Attributes:
	1...	AMDUNQ	The data set has nonunique keys.
	0...		The data set has unique keys.
	.1..	AMDFAULT	The data set is staged by cylinder fault
	.0..		The data set is staged at open time, if required
	..1.	AMDBIND	The data set is staged and bound
	..0.		The data set is not bound
	...1	AMDWAIT	Destaging is completed before control is returned to the program that closes the data set
	...0		There is no waiting to return control
	.... 1...	AMDLM	The data set is loaded.
	.... 0...		The data set is in load mode or is not loaded.
	xxx		Reserved
41 (29)	7		Reserved
48 (30)	48	AMDSTAT	Data set statistics:
48 (30)	8	AMDSTSP	OS/VS system time stamp
56 (38)	2	AMDNIL	Number of index levels
58 (3A)	2	AMDNEXT	Number of extents in the data set
60 (3C)	4	AMDNLR	Number of user-supplied records in the data set
64 (40)	4	AMDDELR	Number of deleted records
68 (44)	4	AMDIREC	Number of inserted records
72 (48)	4	AMDUPR	Number of updated records
76 (4C)	4	AMDRETR	Number of retrieved records
80 (50)	4	AMDASPA	Number of bytes of free space in the data set
84 (54)	4	AMDNCIS	Number of times a control interval was split
88 (58)	4	AMDNCAS	Number of times a control area was split
92 (5C)	4	AMDEXCP	Number of times EXCP was issued by VSAM I/O routines

CAXWA—Catalog Auxiliary Work Area

The CAXWA is built when an OS/VS2 master or user catalog is opened or is being created. The CAXWA is used to contain the addresses of control blocks and work areas needed when a catalog such as the alternate TIOT, the DRWA, and the UCB is being opened or created. The CAXWA also contains flags that indicate the type of processing being performed on the catalog and the OS/VS component that invoked the processing. Each CAXWA points to a catalog's ACB. The ACB describes the catalog as a key-sequenced VSAM data set. All CAXWAs that describe the catalogs available to a user's program are chained together. The AMCBS (CBSCAXCN) contains the address of the CAXWA chain.

Catalog Auxiliary Work Area (CAXWA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1	CAXID	Control block identifier, X'CA'
1 (1)	3		Reserved
4 (4)	4	CAXCHN	Address of the next CAXWA in the chain
8 (8)	1	CAXFLGS	Flags:
	1...	CAXBLD	Build request
	.1..	CAXOPN	The catalog is being opened
	..1.	CAXCLS	The catalog is being closed
	...1	CAXEOV	An End of Volume routine is in control
	.... 1...	CAXCMP	Open/Close/EOV processing is complete
	.... .1..	CAXMCT	Identifies the OS/VS2 master catalog
	.... .0..		Identifies an OS/VS2 user catalog
	10	CAXCMR	Catalog management has been called by another catalog management routine
	01	CAXSCR	Catalog Management (SVC 26) has been called by the OS/VS Scheduler
	00		Catalog management (SVC 26) has been called by an Access Method Services procedure
9 (9)	1	CAXFLG2	Flags:
	1...	CAXF2DT	The catalog has been deleted. The following flags are set by IFG0191X and IFG0200N:
	.1..	CAXF2NDD	No DDNAME found
	..1.	CAXF2NCR	Unable to obtain virtual storage with a GETMAIN request
	...1	CAXF2IOE	I/O error
	.... 1...	CAXF2CLR	RPL cleanup request
	.... .1..	CAXF2CA	If an error occurs, free the CAXWA
	.... ..1.	CAXF2REC	Recoverable catalog
	1	CAXF2VTU	Volume time stamp updated
10 (A)	1	CAXFLG3	Flags:
	1...	CAXF3AT	CRA alternate TIOT exists
	.1..	CAXF3ANE	CRA does not exist
	.... 00..	CAXF3B5	Password not read
	.... 11..		No passwords
	.... 01..	CAXF3B6	Master password; no update password
	.... 10..		Master and update passwords
	...1	CAXRAC	Catalog is RACF-defined
	..x. ..xx		Reserved
11 (B)	1	CAXACT	Catalog activity count
12 (C)	4	CAXATIOT	Address of the alternate TIOT
16 (10)	4	CAXSCHWA	Address of the Scheduler work area
20 (14)	4	CAXDRWP	Address of the catalog's DRWA
24 (18)	4	CAXACB	Address of the catalog's ACB
28 (1C)	4	CAXUCB	Address of the UCB
32 (20)	12	CAXCCR	Catalog control record information
32 (20)	3	CAXHACI	Control interval number of the highest allocated control interval in the catalog

Catalog Auxiliary Work Area (CAXWA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
35 (23)	3	CAXNFCI	Control interval number of the next free control interval in the catalog
38 (26)	3	CAXCDCI	Number of deleted control intervals
41 (29)	3	CAXFDCI	Control interval number of the first deleted control interval in the catalog
44 (2C)	2		Reserved
46 (2E)	2	CAXRPLCT	Number of RPLs associated with the CAXWA
48 (30)	4	CAXRPL	Address of the first RPL in the CAXWA's RPL chain
52 (34)	44	CAXCNAM	Catalog's DSNAME
52 (34)	6	CAXVOLID	CRA volume serial
58 (3A)	4	CAXRACTS	CRA creation time stamp
62 (3D)	4	CAXRATEP	CRA's TIOT entry address
66 (42)	8	CAXRADDN	CRA's DDNAME
74 (4A)	22		Reserved for CRA
96 (60)	4	CAXOPLST	Open/Close parameter list:
96 (60)	1	COPTS	Option flags:
	1...	CENLST	End of list indicator
	.xxx xxxx		Reserved
97 (61)	3	COPACB	Address of the catalog's ACB
100 (64)	4	CAXOPEWA	Address of the Open/Close/EOV work area
104 (68)	4	CAXCCA	Address of the CCA
104 (68)	4	CAXPLOCK	Cross-memory POST lock
108 (6C)	4	CAXJDE	Address of the JDE
112 (70)	4	CAXCRACB	Address of the CRA's ACB
116 (74)	4	CAXRACP	Address of RACF profile
120 (78)	4	CAXECB	RPL WAIT ECB
124 (7C)	4	CAXASCBP	Address of waiting memory's ASCB
128 (80)	2	CAXHRPLC	Hung-up RPL count. A hung-up RPL is one that was not freed because an ABEND occurred while I/O was in process.
130 (82)	2	CAXWNQID	RPL WAIT ENQ count

CCA—Catalog Communications Area

The CCA is built each time an OS/VS component issues the CATLG macro instruction (SVC 26) to process an OS/VS2 master or user catalog record. The CCA contains information about the catalog being processed, and about the catalog record and its extensions contained in each of the six buffers (RABs) available to process the user's request. The CCA is used to pass information between catalog management procedures. Register 11 contains the address of the CCA currently being processed. The CAXWA (CAXCCA) points to the CCA that describes the request currently processing its catalog's records.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	2	CCAID	CCA ID='ACCA'
2 (2)	2	CCASZ	Size of CCA
4 (4)	4	CCAPROB	Problem determination
	2	CCAMODID	Error module ID
6 (6)	2	CCAERRCD	Error codes (DSECT for next two bytes)
	1	CCAREASN	Set reason code
		CCACDR	Refer reason code
	1	CCARETRN	Set return code
		CCACD1	Refer return code
8 (8)	5		Reserved
13 (D)	1	CCACD2	Return code 2
			Note: See "Diagnostic Aids: Error Codes" for a list of the VSAM Catalog Management return codes and error codes.

Catalog Communications Area (CCA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
14 (E)	1	CCAFLG1	Flags Byte 1:
	1...	CCAF1LPS	Stop the loop
	.1..	CCAF1CNS	Master catalog is being searched for a user catalog catalog record
	..1.	CCAF1LRD	CCR read into core
	...1	CCAF1KEY	Retrieve the catalog record based on a DSNAME value
	.... 1...	CCAF1KGE	Retrieve the next catalog record
	.... .1..	CCAF1CR	A checkpoint of the CCR is required
	.... ..1.	CCAF1UP	GET for update
	1	CCAF1DK	When the caller is renaming a data set, this flag indicates that the data set's true-name record is to be deleted, but the data set's catalog record is not to be deleted.
15 (F)	1	CCAFLG2	Flags Byte 2:
	1...	CCAF2SYS	The caller is an OS/VS system module
	.1..	CCAF2NVC	No validity check on the caller's CTGFL or work area is required
	..0.	CCAF2CCT	Search all catalogs available to the caller.
	..1.		Search the first OS/VS2 catalog specified by the caller's STEPCAT statement. If the caller's JCL doesn't include a STEPCAT statement, search the OS/VS2 master catalog. In either case search only one catalog.
	...1	CCAF2XEQ	Exclusive enqueue
	...0		Shared enqueue
	.... 1...	CCAF2RHS	When a catalog management routine calls the VSAM Open routines to open a newly created catalog, and the Open routines call OS/VS2 Catalog Management routines to obtain information about the catalog to be opened, the situation is called a "recursive call". The catalog cannot be dequeued when the Catalog Management routines return to the caller (VSAM Open routines).
	.... .xx.	CCAF2COB	Both catalog open and build:
	.... .1..	CCAF2CO	Catalog is being opened
	.... ..1.	CCAF2CB	Catalog is being created
	1	CCAF2SMO	Search the master catalog only
16 (10)	1	CCAFLG3	Flags Byte 3:
	1...	CCAEXGR1	Exit indicator
	.1..	CCAGC4	The catalog record contains a password information set of fields (identified by type (group) code 4) (detected during IGGPSCNC processing)
	..1.	CCAGDSP	Caller specified the GENDSP option (detected during IGGPSCNC processing)
	...1	CCAEXGR2	Exit indicator
	.... 1...	CCANF	The set of fields cannot be found
	.... .1..	CCAELC2	Exit indicator
	.... ..1.	CCALFT	First time
	1	CCAEGREC	Exit indicator

Catalog Communications Area (CCA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
17 (11)	1	CCAFLG4	Flags Byte 4:
	1...	CCAF4DRQ	The catalog must be dequeued after the request completes
	.1..	CCAF4BYS	Bypass the security verification
	..1.	CCAGVNC	The required variable-length field is not completely contained in the record currently in the buffer
	...1	CCAGVNF	The set of fields identified by the caller-specified sequence number cannot be found
	.... 1...	CCAGVNBS	There is no buffer space available to contain an extension record
	.... .1..	CCAGVEX	Exit indicator
	.... ..1.	CCAGVNE	The field does not exist in the located set of fields
	1	CCATCOMP	Test complete: all set-of-fields pointers have been examined and all designated fields have been tested
18 (12)	1	CCAFLG5	Flags Byte 5:
	1...	CCAMEX2	Exit indicator
	.1..	CCAMEX	Exit indicator
	..1.	CCAMEX1	Exit indicator
	...1	CCAMODPA	The catalog record's base record must be written (using IGGPPAD) into the catalog
	.... 1...	CCATHIT	Successful test: a set of fields has been found that satisfies the test conditions
	.... .1..	CCATEX	Exit indicator
	.... ..1.	CCATEX1	Exit indicator
	1	CCATEX2	Exit indicator
19 (13)	1	CCAFLG6	Flags Byte 6:
	1...	CCAMCODR	The catalog must be dequeued when the request completes
	.1..	CCADELP	A deleted set-of-fields pointer was found
	..1.	CCAMNOSP	The catalog record's free space isn't large enough to contain all the new catalog information during the set of fields move operation
	...1	CCAINIT	Insert switch for variable-length field being retrieved
	.... 1...	CCASUPFD	Suppress password field information during field retrieval
	.... .1..	CCAREUSE	The contents of the caller's record areas (buffers) can be used by IGGPEXT and IGGPMOD
	.... ..1.	CCAEXT	Set when a catalog management routine calls the Extract routine (IGGPEXT)
	1	CCAMOD	Set when a catalog management routine calls the Modify routine (IGGPMOD)
20 (14)	4	CCATCB	Address of the TCB
20 (14)	4	CCALBCYL	Address of the label cylinder data
24 (18)	4	CCARB	Address of the RB
24 (18)	4	CCADPL	Address of the DADSM parameter list
28 (1C)	4	CCACPL	Address of the caller's CTGPL
32 (20)	4	CCAACB	Address of the catalog's ACB
36 (24)	4	CCANPCCB	Address of the next PCCB

Offset	Bytes and Bit Pattern	Field Name	Description
40 (28)	4	CCAURAB	Address of the record area block (RAB) currently in use
44 (2C)	44	CCASRCH	Search argument (DSNAME of a cluster, data set, index, catalog, alias, generation data group, or nonVSAM data set, or a volume serial number)
44 (2C)	3	CCASRID CCASRCIN	Control interval number
47 (2F)	41		Remainder of CCASRCH
88 (58)	20	CCARAB0	Record Area Block 0: Each record area block describes the catalog record contained in one of the six catalog management buffers available for the request. RABs 1 through 5 are identical in format to RAB 0. **Note:** 'x' in each field name is replaced by '0' through '5' to indicate a particular RAB's field.
88 (58)	1	CCARxFLG	Flags: The following flag is used by IGGPEXT and IGGPMOD:
	1...	CCARxUR	The RAB is in use. It cannot be used by IGGPEXT or IGGPMOD
	.1..	CCARxU1	The RAB is temporarily in use by IGGPEXT or IGGPMOD. It cannot be overlaid.
	..1.	CCARxU2	(Same as CCARxU1)
	...1	CCARxWR	The buffer must be written before another catalog record can be read into it
	.... 1...	CCARxPA	The buffer contains a new catalog record—PUT-ADD is required to add the record to the catalog
	.... .xx.		Reserved
	1	CCARxUPD	The buffer contains a GET-for-update record
89 (59)	3	CCARxRPL	Number of the last assigned RPL in the RPL index
90 (5A)	2		Reserved
92 (5C)	4	CCARxREC	Address of the record in the buffer
96 (60)	12	CCARxSEG	Addresses of parts of the catalog record:
96 (60)	4	CCACPE2x	Address of the first byte after the fixed-length header fields.
100 (64)	4	CCACPE3x	Address of the first set of fields
104 (68)	4	CCACPE4x	Address of the first free-space byte in the record
108 (6C)	20	CCARAB1	Record Area Block 1 (See RAB 0 description)
128 (80)	20	CCARAB2	Record Area Block 2 (See RAB 0 description)
148 (94)	20	CCARAB3	Record Area Block 3 (See RAB 0 description)
168 (A8)	20	CCARAB4	Record Area Block 4 (See RAB 0 description)
188 (BC)	20	CCARAB5	Record Area Block 5 (See RAB 0 description)

Catalog Communications Area (CCA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
208 (D0)	1	CCARPLK	Work area for IGG0CLAG and IGG0CLAM
209 (D1)	1	CCARPLF	Work area for IGG0CLAG and IGG0CLAM
210 (D2)	1	CCARPLX	Work area for IGG0CLAG and IGG0CLAM
211 (D3)	1	CCARPLT	Work area for IGG0CLAG and IGG0CLAM
212 (D4)	6		Reserved
218 (DA)	1	CCASC	Code to indicate the type of search being performed
219 (DB)	1	CCAQLEN	Length of the 1st-level qualifier name (0 if the data set's dsname is not qualified).
220 (DC)	4	CCARPL1	Address of the RPL in use
224 (E0)	44	CCADESA	Save area for the extent information returned by OS/VS DADSM and OS/VS2 Catalog Management: Suballocate
224 (E0)	1	CCANDEXT	Number of extents
225 (E1)	1	CCAIXEXT	Extent index value
226 (E2)	2	CCASSVOL	Sequence number of the data set directory entry in the volume catalog record
228 (E4)	40	CCAEXTDE	Five 8-byte extent descriptors:
	2	CCAEXTSS	Sequence number of the Data Space Group set of fields that this extent's space is a part of.
	4	CCAEXTAD	The extent's starting physical address:
	2	CCAEXTCC	Cylinder number CC
	2	CCAEXTHH	Head number HH
	2	CCAEXTTH	Number of tracks in the extent
268 (10C)	1	CCAASCIK	Number of control intervals required to satisfy the caller's request
269 (10D)	1	CCACRRP	RPL used to read CCR
270 (10E)	1	CCAASCIX	Used by the ASSIGN functions—points to the element in CCAASCI currently being processed
271 (10F)	9	CCAASCI	Number of each assigned control interval
280 (118)	16	CCAEQDQ	ENQ/DEQ parameter list
280 (118)	1	CCAEDXFF	"End of parameter list" indicator—X'FF'
281 (119)	1	CCAEDRLN	Length of minor name
282 (11A)	1	CCAEDOPT	ENQ/DEQ options
	1...	CCAEDSHR	Shared
	0...		Exclusive
	.xxx xxxx		Reserved
283 (11B)	1	CCAEDRCD	ENQ/DEQ return code
284 (11C)	4	CCAEDQNM	Address of the major name
288 (120)	4	CCAEDRNM	Address of the minor name
292 (124)	4	CCAEDUCB	Address of the UCB
296 (128)	4	CCAMLRET	Address of the caller's save area
300 (12C)	12	CCAMSSPL	Storage management work area

Catalog Communications Area (CCA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
300 (12C)	4	CCAMNLLP	Number of bytes to process
304 (130)	4	CCAMNPTR	Address of the return address
308 (134)	1		Storage management byte
309 (135)	1	CCAMNSPL	Required subpool number
310 (136)	2		Storage management byte
312 (138)	4	CCARPRM	Return parameters
316 (13C)	8	CCACMS	Catalog Management Services work area
316 (13C)	4	CCACMSWA	Address of the Catalog Management Services calling routine's work area
320 (140)	4	CCAEXCMS	Address of a secondary Catalog Management Services work area

The following fields are set and used by IGGPLOC, IGGPEXT, IGGPMOD, and IGGPTSTS, and catalog management subfunctions that these procedures call:

Offset	Bytes and Bit Pattern	Field Name	Description
324 (144)		CCALUME	Field management work area:
324 (144)	4	CCACPE5	Address of a selected set-of-fields pointer
328 (148)	4	CCACPE51	Address of a second selected set-of-fields pointer
332 (14C)	4	CCACPE52	Address of a third selected set-of-fields pointer
336 (150)	4	CCACPE53	Address of a fourth selected set-of-fields pointer
340 (154)	4	CCACPE6	Address of a selected set of fields
344 (158)	4	CCACPE61 CCARABSE	Address of a second selected set of fields
348 (15C)	4	CCACPE7 CCAIDPT	Address of field value
352 (160)	4	CCACPE71	Address of a second selected retrieved field
356 (164)	2	CCAGOPLN	Length of the set-of-fields pointer
358 (166)	2	CCASL	Number of bytes for the sequence number
360 (168)	4	CCAILNG	Length of the selected retrieved field
364 (16C)	4	CCAFLPT CCATFLPT	Address of the requested-field CTGFL Address of the CTGFL-for-tests
368 (170)	4	CCARABPT	Address of the record area block
372 (174)	4	CCADICT	Dictionary information to describe the field, based on its field name
376 (178)	4	CCAXCPL CCAMCPL	Address of the CTGPL built when IGGPEXT and IGGPMOD are called, so that information in the caller's CTGPL is not altered
380 (17C)	4	CCARABB	Address of the RAB that identifies the base catalog record
384 (180)	4	CCARABF	Address of the RAB that identifies the first record area (buffer) that can be used by IGGPEXT or IGGPMOD

Offset	Bytes and Bit Pattern	Field Name	Description
388 (184)	4	CCARABL	Address of the RAB that identifies the last record area (buffer) that can be used by IGGPEXT or IGGPMOD
392 (188)	3	CCACBASE	The control interval number of the base catalog record
395 (18B)	1	CCAGC	Group code of the requested set of fields
396 (18C)	2	CCALREL CCALREL1	Relative repetition number of a selected set of fields
398 (18E)	2	CCASN CCASN1	Sequence number of a selected set of fields
400 (190)	2		Reserved
402 (192)	2	CCAIXFPL	Index to the current CTGFL being processed
404 (194)	2	CCAIXREL	Index for CCATREL
406 (196)	2	CCATNREL	The sequence number of the next set of fields to perform tests against if CCATREL is full or if there are no buffers available to contain the catalog record's next extension
408 (198)	2	CCATNUM	Number of successful relative repetition numbers (cannot exceed 16)
410 (19A)	32	CCATREL	Successful relative repetition numbers
442 (1BA)	2	CCATNO	Total number of successful relative repetition numbers (might exceed 16)
444 (1BC)	4	CCATEST	Address of the test CTGFL
448 (1C0)	20	CCARBA	Work area for extent descriptors
448 (1C0)	6	CCACRAVL	CRA's volume serial
448 (1C0)	2	CCASS	Sequence number of the Data Space Group set of fields that contains the extent
450 (1C2)	4	CCACCHH1	Physical address—CCHH—of the extent's first track
454 (1C6)	4	CCACCHH2	Physical address—CCHH—of the extent's last track
454 (1C6)	4	CCACRADT	CRA device type
458 (1CA)	2	CCATT	Number of tracks in the extent
460 (1CC)	4	CCARBA1	Low relative byte address (RBA)
464 (1C0)	4	CCARBA2	High relative byte address (RBA)
468 (1D4)	2	CCATLNG CCATLEN	The total length of the extent information that has been processed
470 (1D6)	2	CCARBAL	RBA extent balance
472 (1D8)	2	CCACNIX	Combination name index
474 (1DA)	2	CCASMFIX	The type of SMF record to be written when a cluster or catalog is defined (SMF record for the cluster, data set, or index)
476 (1DC)	4	CCAIDPT2	Address of the available space in the caller's work area or of the caller-supplied update information
480 (1E0)	4	CCAIDPT3 CCARABSM	Address of the length-field of a variable-length field in the user's RAB

Catalog Communications Area (CCA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
484 (1E4)	2	CCAGVCT	Number of set-of-fields pointers processed so far
486 (1E6)	2	CCANEVV	If the requested variable-length field is non-existant, this field is set to binary zero
488 (1E8)	3	CCAGVEXT	Control interval number of the record's next extension record (not yet in a buffer)
491 (1EB)	1	CCANEFV	If the requested fixed-length field is non–existant, this byte is set to X'FF'
492 (1EC)	1		Reserved
493 (1ED)	1	CCAGRGC	Type (group) code of the requested set of fields
		CCARCDID	Record ID
494 (1EE)	2	CCAGRHI CCAGRHI1	High sequence number of the requested set of fields
		CCASSEQ	Sequence number
496 (1F0)	2	CCAIXTPL	Index to test CTGPLs
498 (1F2)	2	CCADLEN	Number of bytes to be deleted from the catalog record
500 (1F4)	2	CCADIFF	The difference between the insert length and the delete length (can be a negative number)
502 (1F6)	2	CCAREPCT	Number of relative repetition numbers processed so far
504 (1F8)	2	CCADISP	Displacement into variable-length field to the delete/insert location
506 (1FA)	3	CCASVCI	Save area for the control interval number of the base catalog record
509 (1FD)	3	CCASVCI1	Save area for the control interval number
512 (200)	4	CCADTA	Address of the dictionary
516 (204)	4	CCACDTA	Address of the index combination table
520 (208)	2	CCADTCT	Number of dictionary entries
522 (20A)	2	CCACDTCT	Number of index combination entries
524 (20C)	4	CCACWAP	Address of the controller work area
528 (210)	4	CCASDWAP	Address of the virtual storage obtained by a GETMAIN request
532 (214)	4	CCAILNG3	Save area for the insertion length
536 (218)	4	CCAILNG2	Length of the user-supplied insert data
540 (21C)	4	CCAALPTR	Address of the space management work area
544 (220)	4	CCASMFPT	Address of the SMF record chain
548 (224)	4	CCALCPL	Address of the CTGPL used when a catalog management routine issues the LSPACE macro instruction

Offset	Bytes and Bit Pattern	Field Name	Description
552 (228)	1	CCAFLG7	Flag byte 7:
	1...	CCALSP	LSPACE has been called by a catalog management routine
	.1..	CCASMFEX	SMF exit indicator
	..1.	CCASMFA	Perform SMF ALTER processing during IGGPEXT and IGGPMOD processing
	...1	CCASMFBR	The base catalog record is in a record area (buffer) for SMF processing
	.... 1...	CCAONCE	Move only one set of fields
	.... .1..	CCAROREQ	Request-type is read-only
	.... ..1.	CCAFEOV	Force end-of-volume
	1	CCACRABU	CRA being built
553 (229)	1	CCAFLG8	Flags byte 8:
	1...	CCADSRCL	Recursive call for DEFINE space
	.1..	CCAVBUFI	Number of volume records buffered
	..1.	CCASCRA	Suppress CRA updates
	...1	CCASCICK	Suppress CRA control interval check
	.... 1...	CCALPIND	Loop control in buffer scan for GETs
	.... .1..	CCAVRIND	Volume record buffer chain is to be checked
	.... ..1.	CCALEOD	End-of-file on low keys
	1	CCAAUCAT	Volume has a user catalog
554 (22A)	1	CCAFLG9	Flags byte 9:
	1...	CCARABYC	Bypass catalog I/O
	.1..	CCARAEOV	CRA end-of-volume
	..1.	CCARALRD	CRA CCR has been read
	...1	CCARACR	CRA CCR checkpoint requested
	.... 1...	CCAUCRA	Use CRA CI number translate table before CRA GET
	.... .1..	CCARAACT	CRA is active
	.... ..1.	CCARAICI	Inhibit catalog I/O
	1	CCARESUM	Replace sum
	0		Increment sum
555 (22B)	1	CCANORBA	Number of RBAs needed in control block structure
556 (22C)	4	CCASMFRD	Address of the SMF record
560 (230)	2	CCASMFCT	Number of catalog records in the SMF record chain to be used in creating the SMF ALTER record
562 (232)	2	CCASMFLG	SMF record flags:
562 (232)	1	CCASMFG1	First flag byte
	Before calling the SMF-record-writing routine, the caller sets these bits to indicate what he has done or intends to do:		
	1...	CCASMFUC	Uncatalog—write SMF record type 67
		CCASMFDF	DEFINE—write SMF record type 63
	.1..	CCASMFSR	Data set, index, or nonVSAM data set has been scratched—write SMF record type 67
		CCASMFAL	ALTER—write SMF record type 63
	..xx xxxx		Reserved
563 (233)	1	CCASMFG2	Reserved
564 (234)	2	CCASMFLN	Amount of virtual storage obtained for the SMF record
566 (236)	3	CCACHAIN	Control interval number save area
569 (239)	3	CCACI1	Control interval number save area

Offset	Bytes and Bit Pattern	Field Name	Description
572 (23C)	3	CCACI2	A second control interval number save area
575 (23F)	3	CCACI3	A third control interval number save area
578 (242)	2	CCAVARLN	Number of bytes to be inserted into the record
580 (244)	4	CCARRAB	Address of the RAB containing the set-of-fields pointers where delete/insert processing is to begin.
584 (248)	4	CCARBASE	Address of a second RAB, similar to CCARRAB
588 (24C)	4	CCAVARPT	Address of the information to be inserted into the record
592 (250)	2	CCADELN	Number of bytes to be deleted from the record
594 (252)	20	CCAVAR	Insert information save area
614 (266)	20	CCAVAR1	A second insert information save area
634 (27A)	3	CCADEL1	The control interval number of the first record in a series of records to be deleted
637 (27D)	3	CCADEL2	The control interval number of the last record in a series of records to be deleted
640 (280)	40	CCAXLATE	Translation work area
680 (2A8)	4	CCAR14S	IGG0CLC9 Register 14 save area
684 (2AC)	8	CCABMINP	Input parameters for the bit manipulation routine (IGG0CLBR)
684 (2AC)	2	CCABMTRK	Starting track number
686 (2AE)	2	CCABMLIM	Ending track number, or upper limit
688 (2B0)	2	CCABMMIN	Minimum number of tracks required
690 (2B2)	1	CCABMFLG	State and function code:
	1...	CCABMST	State to set or condition to check
	.1..	CCABMCHK	Perform the check
	..1.	CCABMSET	Set the state
	...1	CCABMCCK	Perform a conditional check
	.... 1...	CCABMLST	Last set required
	xxx		Reserved
691 (2B3)	1		Reserved
692 (2B4)	5	CCABMOUT	Output parameters for the bit manipulation routine (IGG0CLBR)
692 (2B4)	2	CCABMONN	Number of tracks
694 (2B6)	2	CCABMOTR	Starting track number
696 (2B8)	1	CCABMOFG	Output flags:
	1...	CCABMOST	State of bits
	.xxx xxxx		Reserved
697 (2B9)	2		Reserved
699 (2BB)	1	CCABMPAD	Padding character
700 (2BC)	4	CCABMGOP	Address of the current space map set of fields
704 (2C0)	4	CCABMPTR	Address of the current bit mask byte

Catalog Communications Area (CCA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
708 (2C4)	4	CCABMEND	Address of the end of the current space map set of fields
712 (2C8)	2	CCABMBT1	Number of bits—first byte
714 (2CA)	2	CCABMBT	Number of bits—last byte
716 (2CC)	2	CCABMBYT	Number of full bytes
718 (2CE)	2	CCABMSTR	Current bit mask starting track
720 (2D0)	4	CCABMWK1	Work area
724 (2D4)	4	CCABMWK2	Work area
728 (2D8)	4	CCABMWK3	Work area
732 (2DC)	4	CCABMWK4	Work area
736 (2E0)	4	CCABMRB1	Address of the first RAB that points to the space map set of fields
740 (2E4)	4	CCABMRB2	Address of the second RAB that points to the space map set of fields
744 (2E8)	40	CCATEMPS	Compiler temporary work area
784 (310)	256	CCAMNCAT	Contiguous area for tracking
784 (310)	248	CCAMNAT	Tracking buffer
784 (310)	1	MNATTOP	First byte in the buffer—flag byte
	1...	MNATFULL	Buffer is full
	.xxx xxxx		Reserved
785 (311)	239		Remainder of the buffer
1024 (400)	1	MNATFLGS	Flags for the most recent entry in the buffer
	1...	MNATVAL	Valid entry
	1	MNATSCLS	Class 'S' virtual storage
	.xxx xxx.		Reserved
1025 (401)	3	MNATARG1	Remainder of the first word of the most recent entry
1028 (404)	4	MNATARG2	Second word of the most recent entry
1032 (408)	1	CCAMNLL	GETMAIN/FREEMAIN length list—end-of-list byte
1033 (409)	3	CCAMNLEN	GETMAIN/FREEMAIN length
1036 (40C)	4	CCAMNADR	GETMAIN/FREEMAIN address
1040 (410)	16	CCAARFWA	Spill routine work area
1040 (410)	4	ARFGMLEN	GETMAIN length list
1040 (410)	1	ARFGMLP	End-of-list byte
1041 (411)	3	ARFLEN	Length of GETMAIN
1044 (414)	4	ARFGMADR	Address for GETMAIN
1048 (418)	4	ARFSBSCH	Address of first spill block
1052 (41C)	4	ARFSBECH	Address of last spill block

Catalog Communications Area (CCA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
1056 (420)	1	CCARVFG1	Recovery flags:
	1...	RVCCAV	CCA is valid
	.1..	RVARFI	Tracking data is incomplete
	..1.	RVCMSFG	Catalog management services function gate
	...1	RVESBO	ESTAE backout is in control
	.... 1...	RVESBOR	ESTAE backout register
	.... .1..	RVRPLMFG	RPL management gate
	.... ..1.	RVWG	RPL WAIT gate
	x		Reserved
1057 (421)	3		Reserved
1060 (424)	348	CCAREGS	Register 12, 13, and 14 save area (See "Diagnostic Aids" for details about the CCA register save area)
1060 (424)	4		Address of the user save area
1064 (428)	8	CCAMODNM	Load module name
1408 (580)	4	CCABZSAV CCADSPWA	Address of the save area for IGG0CLBZ Address of DEFINE SPACE work area
1412 (584)	4	CCACUMPL	Address of the catalog management upgrade parameter list
1416 (588)	1		Reserved
1417 (589)	3	CCASBASE	Save base control interval for upgrade process
1420 (58C)	4	CCACRACI	Address of the CRA record pointer array
1424 (590)	4	CCARAACB	Address of the CRA ACB
1428 (594)	4	CCARARPL	Address of the CRA RPL
1432 (598)	4	CCARARBA	Relative byte address of the CRA
1436 (59C)	4	CCARAREC	Address of a CRA record
1440 (5A0)	4 2	CCARALSA CCACRABT	Address of the CRA local save area Block/track value for CRA record construction
1442 (5A2)	2		Reserved
1444 (5A4)	1	CCAFLG10	Flag byte 10:
	1...	CCAINCPL	Catalog parameter list is invalid
	.1..	CCAPDMW	Problem determination message
	..1.	CCACATAC	Catalog is active
	...1	CCARAFEV	Forced end of volume
	.... 1...	CCARARTC	Recovery exit—return to caller
	xxx		Reserved
1445 (5A5)	3	CCASUMTT	Sum of the tracks in the CRA
1448 (5A8)	4	CCADICTS	Data/index time stamp
1452 (5AC)	8	CCARANCA	Start, end addresses of normal record buffer chain
1460 (5B4)	8	CCARAVCA	Start, end addresses of volume record buffer chain
1468 (5BC)	8	CCAVTS	Volume time stamp
1476 (5C4)	4	CCAREWKA	Address of reusable data set processing work area
1480 (5C8)	4	CCASMFP	Save area for SMF problem determination
1480 (5C8)	2	CCASMFMD	Module ID

Catalog Communications Area (CCA)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
1482 (5CA)	1	CCASMFRC	Reason code
1483 (5CB)	1	CCASMFCD	Return code
1484 (5CC)	4	CCAPROBX	Save area for CCAPROB
1484 (5CC)	2	CCAMODDX	Error module ID
1486 (5CE)	2	CCAERCDX	Error codes
1486 (5CE)	1	CCARESNX	Reason code
1487 (5CF)	1	CCARETRX	Reason code
1488 (5D0)	4	CCADGDGA	Address of generation data group work area
1592 (dD4)	12	CCAREQDQ	RPL ENQ/DEQ parameter list
1604 (5E0)	4		Reserved

CTGCV—Catalog Control Volume List

The CTGCV is built by the issuer (primarily the Scheduler) of a Superlocate request by data set name for volume information. Catalog management uses the CTGCV to return to the caller the name of the catalog that contains the data set name. The CTGPL points to the CTGCV.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	6	CTGCVVOL	Volume serial number of the catalog named in CTGCVDSN.
6 (6)	44	CTGCVDSN	Name of the catalog in which the data set name was located, or which needs to be opened to continue the superlocate.
6 (6)	4	CTGCVDEV	CVOL device type
50 (32)	44	CTGCVCC	An alias of the catalog (catalog connector). Returned only when the catalog is identified by the first qualifier of the data set name and the first qualifier is an alias of the catalog name.

CTGFL—Field Parameter List

The CTGFL is built before an OS/VS component issues the CATLG macro instruction (SVC 26) to process an OS/VS2 user or master catalog record. The CTGFL defines one of the catalog record's fields or a group of logically related fields (identified by a combination name). The CTGFL is used in two situations:

- It identifies a catalog record field to retrieve or update. The CTGPL contains the address of each CTGFL used in this way.
- It identifies a catalog record field to compare against caller-supplied data. This is a "test" CTGFL and is addressed by another CTGFL.

For UPDATE-Extend processing, one or three CTGFLs identify the volume information set(s) of fields to be extended. The catalog record fields identified by the CTGFL(s) are not explicitly retrieved or updated for the caller.

When a catalog management routine is processing a CTGFL, the CTGFL's address is in the CCA (CCAFLPT or CCATEST).

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1	CTGFLDNO	Number of entries in CTGFLDAT
1 (1)	1	CTGFLDCD	Test condition:
	X'00'		The CTGFL describes a field to be updated or retrieved. The CTGFL is pointed to by the caller's CTGPL (CTGFIELD entry).
	Not X'00'		The CTGFL describes a test condition. This CTGFL is pointed to by another CTGFL.
			Test condition:
	X'80'	CTGFLDEQ	Equal
	X'60'	CTGFLDNE	Not equal
	X'20'	CTGFLDGT	Greater than
	X'40'	CTGFLDLT	Less than
	X'A0'	CTGFLDGE	Greater than or equal
	X'C0'	CTGFLDLE	Less than or equal
	X'80'	CTGFLDZ	Test under mask for zeros
	X'10'	CTGFLDON	Test under mask for ones
	X'40'	CTGFLDMX	Test under mask for mixed
2 (2)	1	CTGFLDGC	Type (group) code
3 (3)	1	CTGFLDRE	Test results:
	xxxx xxx.		Reserved
	0	CTGFLDTS	Successful test
	1		Test failed
4 (4)	4	CTGFLDWA	Work area: contains information about the catalog record's field name from the dictionary
8 (8)	4	CTGFLDNM	Address of the field name
12 (C)	4	CTGFLCHN	Address of a CTGFL-for-tests, or 0
16 (10)	VL	CTGFLDAT	Address and length, in the caller's work area, of: • Each field that was retrieved, if the request was LOCATE or LISTCAT. • New data to replace or add to data in the catalog record, if the request was UPDATE, DEFINE, or ALTER. • Data used to compare to catalog record fields, if the CTGFL is a CTGFL-for-tests. The length of the CTGFLDAT field is the CTGFLDNO value times 8 (two 4-byte fields):
		CTGFLNG	4-byte length of the data
		CTGFLPT	4-byte address of the data

CTGFV—Field Vector Table

The CTGFV is built by the Access Method Services utility programs and contains addresses of user-supplied information fields and lists. The CTGFV is built when the user issues a DEFINE or ALTER command. If the user is creating a cluster, a CTGFV is built for each catalog record that will be built to describe the cluster: that is, Access Method Services DEFINE processing builds a cluster CTGFV, a data CTGFV, and, if the cluster is key-sequenced, an index CTGFV. The CTGFV is pointed to by the CTGPL (CTGFVT). If Access Method Services builds more than one CTGFV, the cluster CTGFV is pointed to by the CTGPL (CTGFVT) and the data and index CTGFVs are pointed to by the cluster CTGFV.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1	CTGFVTYP	The CTGFV contains information used by the DEFINE routines to build a catalog record of the type:
	C'A'	CTGFVALN	NonVSAM data set
	C'B'	CTGFVGBS	Generation data group (GDG) base
	C'C'	CTGFVCL	Cluster
	C'D'	CTGFVDTA	Data set
	C'G'	CTGFVAIX	Alternate index
	C'I'	CTGFVIDX	Index
	C'R'	CTGFVPTH	Path
	C'V'	CTGFVVOL	Volume
	C'X'	CTGFVANM	Alias name
1 (1)	1	CTGFVPRO	Catalog Management Services processing option flags:
	1...	CTGFVAVL	ALTER: Add volumes
	.1..	CTGFVRVL	ALTER: Remove volumes
	..1.	CTGFVNDC	No device-type conversion
	...1	CTGFVDRC	DEFINE a recoverable catalog
	.... 1...	CTGFVRON	Turn RACF indicator on
	.... .1..	CTGFVROF	Turn RACF indicator off
	xx		Reserved
2 (2)	1	CTGFVELM	Element number of CMSPCATR
3 (3)	1		Reserved
4 (4)	4	CTGFVDCH	Address of the cluster's data CTGFV
8 (8)	4	CTGFVICH	Address of the cluster's index CTGFV
12 (C)	4	CTGFVVCH	Address of the space CTGFV
16 (10)	4	CTGFVIND	Address of the associated JCL DD statement
20 (14)	4	CTGFVENT	Address of the entry name CTGFL
24 (18)	4	CTGFVSTY	Address of the security information CTGFL (passwords, codeword, and number of tries)
28 (1C)	4	CTGFVOWN	Address of the owner identification CTGFL
32 (20)	4	CTGFVEXP	Address of the expiration date CTGFL
36 (24)	4	CTGFVCRE	Address of the creation date CTGFL
40 (28)	4	CTGFVVLT	Address of the volume serial number list
44 (2C)	4	CTGFVRNG	Address of the key range list
48 (30)	4	CTGFVDVT	Address of the device type CTGFL (for DEFINE NONVSAM only)

Field Vector Table (CTGFV)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
52 (34)	4	CTGFVSPC	Address of the space allocation information CTGFL
		CTGFVTTR	Address of the DSCB's TTR CTGFL (for DEFINE NONVSAM only)
56 (38)	4	CTGFVAMD	Address of the AMDSB CTGFL (for DEFINE CATALOG and DEFINE CLUSTER only)
		CTGFVFSN	Address of the file sequence number CTGFL (for DEFINE NONVSAM only)
60 (3C)	4	CTGFVATR	Address of the data set attributes CTGFL
64 (40)	4	CTGFVBUF	Address of the buffer size CTGFL
68 (44)	4	CTGFVLRS	Address of the average record size CTGFL
72 (48)	4	CTGFVLMT	Address of the GDG limit CTGFL (for DEFINE GDG BASE only)
		CTGFVEXT	Address of exception exit parameter list
76 (4C)	4	CTGFVGAT	Address of the GDG attributes CTGFL (for DEFINE GDG BASE only)
		CTGFVUPG	Address of RGATTR parameter list
80 (50)	4	CTGFVNAM	Address of the true name CTGFL (for DEFINE ALIAS only)
84 (54)	4	CTGFVPWD	Address of the related object's password
88 (58)	4	CTGFVWKA	Address of the CRA feedback area

CTGPL—Catalog Parameter List

The CTGPL is built before an OS/VS component issues the CATLG macro instruction (SVC 26) to process an OS/VS2 master or user catalog record. The CTGPL defines the catalog management request and its options, the catalog record to be processed, and the OS/VS2 catalog that contains the record. The CTGPL is pointed to by register 1. When the catalog management routines build a CCA to support the request, the address of the CTGPL is put into the CCA (CCACPL).

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	1	CTGOPTN1	First option byte:
	1...	CTGBYPSS	Bypass the catalog management security verification processing
	.1..	CTGMAST	Check the master password
	..1.	CTGCI	Check the control interval password
	...1	CTGUPD	Check the update password
	.... 1...	CTGREAD	Check the read password
	.... .1..	CTGNAME	The CTGENT field contains the address of a 44-byte DSNAME, or a 6-byte volume serial number (padded with binary 0s)
	.... .0..		The CTGENT field contains the address of a 3-byte control interval number
	.... ..1.	CTGCNAME	The CTGCAT field contains the address of a catalog's 44-byte DSNAME.
	.... ..0.		The CTGCAT field contains the address of a 4-byte field containing a VSAM catalog's ACB address
	1	CTGGENLD	Generic locate request

Offset	Bytes and Bit Pattern	Field Name	Description
1 (1)	1	CTGOPTN2	Second option byte:
	1...	CTGEXT	Extend option (with UPDATE)
		CTGNSVS	Catalog cleanup request
	.1..	CTGERASE	Erase option (with DELETE)
		CTGSMF	Write SMF record option (with LSPACE)
		CTGREL	Reset to empty status
		CTGGTALL	Search all catalogs (with LISTCAT)
	..1.	CTGPURG	Purge option (with DELETE)
		CTGVMNT	The caller is VSAM Open/Close/EOV: Volume Mount and Verify routine (IDA0192V)
		CTGRCATN	Return the catalog name (with generic LOCATE)
	...1	CTGGTNXT	Get-next option (with LISTCAT)
	.... 1...	CTGDISC	Disconnect option (with EXPORT)
	.... .1..	CTGOVRID	Erase override option (with DELETE)
	.... ..1.	CTGSCR	Scratch space option (with DELETE NonVSAM)
	x		Reserved
2 (2)	1	CTGOPTN3	Third option byte:
	xxx.	CTGFUNC	Specifies the caller-requested function:
	001.	CTGLOC	LOCATE
	010.	CTGLSP	LSPACE
	011.	CTGUPDAT	UPDATE
	100.	CTGCMS	A Catalog Management Services function (see CTGOPTNS)
	...1	CTGSUPLT	SUPERLOCATE function9
	.... 1...	CTGGDGL	GDG locate request—the caller supplied the base generation level (CTGWAGB field in CTGWA).
	.... .1..	CTGSRH	Search the master catalog only.
	.... .0..		Search the user's catalog first (specified by CTGCAT or, if CTGCAT = 0, search the user's catalogs available to the caller via JOBCAT or STEPCAT DD statements, then search the master catalog)
	.... ..1.	CTGNUM	1=Search one catalog; 0=Search both
	1	CTGAM0	The call is an OS/VS2 catalog management request
	0		The call is an OS catalog management request; the caller supplied a CAMLST parameter list that was translated into this CTGPL and CTGFLs.
3 (3)	1	CTGOPTN4	Fourth option byte:
	1...	CTGLBASE	Locate the base level (with SUPERLOCATE-GDG only)
	.0..	CTGDOCAT	If the needed catalog is not open, dynamically allocate and open it.
	.1..		Do not dynamically open the needed catalog.
	..1.	CTGNPROF	No RACF profile to be defined or deleted.
	...x xxxx		Reserved
4 (4)	4	CTGENT	Address of the catalog record identifier, as defined in CTGOPTN1. When the request is generic locate, byte 1 of the field pointed to by CTGENT is a length byte, followed by a 1 to 43 character generic name.
		CTGFVT	Address of the caller's CTGFV

Offset	Bytes and Bit Pattern	Field Name	Description
8 (8)	4	CTGCAT	Address of the catalog's DSNAME or of a 4-byte field that contains the address of the catalog's ACB, as specified in CTGOPTN1.
		CTGCVOL	Address of an OS/VS system-catalog catalog name area, if the request is SUPERLOCATE. The catalog name area contains the catalog's DSNAME and, if the catalog is identified with an alternate DSNAME, the catalog's alias. The OS/VS2 Scheduler uses this information to build the catalog's PCCB.
12 (C)	4	CTGWKA	Address of the caller's work area
16 (10)	2	CTGDSORG	Data set organization, if the request is SUPERLOCATE
16 (10)	1	CTGOPTNS	Catalog Management Services request options:
	0000 1...	CTGDEFIN	DEFINE
	0001 0...	CTGALTER	ALTER
	0001 1...	CTGDELET	DELETE
	0010 0...	CTGLTCAT	LISTCAT
	0011 0...	CTGCNVTV	CONVERT
	xxx		Reserved
17 (11)	1		Reserved
18 (12)	1	CTGTYPE	Type of catalog record:
	C'A'	CTGTALIN	NonVSAM data set
	C'B'	CTGTGBS	Generation data group (GDG) base
	C'C'	CTGTCL	Cluster
	C'D'	CTGTDATA	Data set
	C'F'	CTGTFREE	Free
	C'G'	CTGTAIX	Alternate index
	C'I'	CTGTINDX	Index
	C'M'	CTGTMCAT	Master catalog
	C'P'	CTGTPGSP	Page space
	C'R'	CTGTPTH	Path
	C'U'	CTGTUCAT	User catalog
	C'V'	CTGTVOL	Volume
	C'X'	CTGTANM	Alias name
	C'Y'	CTGTUPG	Upgrade
19 (13)	1	CTGNOFLD	Number of entries contained in CTGFIELD
20 (14)	4	CTGDDNM	Address of the JCL DD statement, if one is associated with this request
		CTGNEWNM	Address of the new DSNAME, if the request is ALTER and the object's name is being changed
If the request is SUPERLOCATE:			
20 (14)	2	CTGFDBK	Feedback area
22 (16)	1	CTGFBFLG	Flags:
	1...	CTGPAR	Parallel mount
	.1..	CTGKEEP	Forced keep
	..1.	CTGGDGB	GDG Base located
	...1	CTGNGDSN	Generation data set name was generated (in the form 'dsname.gxxxxvyy')
	 xxxx		Reserved
23 (17)	1		Reserved

Catalog Parameter List (CTGPL)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
24 (18)	4	CTGJSCB CTGPSWD	Address of the JSCB Address of the caller-supplied password
28 (1C)	VL	CTGFIELD	The 4-byte address of each CTGFL, to specify each catalog field to be processed. The length of CTGFIELD is the CTGNOFLD value times 4.

CTGVL—Catalog Volume List

The CTGVL is built by the issuer of a locate request for a data-set name. Catalog management uses the CTGVL to return to the caller the volume serial numbers of the volumes on which space is allocated to the data set. For superlocate requests, the CTGWA points to the CTGVL.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	6	CTGVLVOL	Volume serial number.
6 (6)	4	CTGVLDEV	Device type.
10 (A)	2	CTGVLSEQ	File sequence number.
12 (C)		CTGVLX	Volume list extension.
12 (C)	3	CTGVLTTR	For a single-volume data set, the TTR of its DSCB.
12 (C)	VL		For a multi-volume data set, a repetition of CTGVLVOL, CTGVLDEV, and CTGVLSEQ for the rest of the volumes.

CTGWA—Catalog Work Area

The CTGWA is built by the caller of catalog management for most requests. The CTGPL points to the CTGWA. The work area has one format for a superlocate request and another format for all other requests.

Offset	Bytes and Bit Pattern	Field Name	Description
Format for a Request other than Superlocate			
0 (0)	2	CTGWALNG	Length of the work area.
2 (2)	VL	*	Data returned for DEFINE, DELETE, GENDSP, LISTCAT, LOCATE, LSPACE.
Format for a Superlocate Request			
0 (0)	4	CTGWAVL	Address of the CTGVL (volume list).
4 (4)	2	CTGWALV	Length of the volume list.
6 (6)	2	CTGWAVCT	Number of volume serial numbers returned in the volume list.
8 (8)	2	CTGWAUCT	Minimum number of volumes that must be mounted.
The following two fields are for a GDG Base only:			
10 (A)	2	CTGWAGCT	CTGGDGB in the CTGPL is set on, and CTGWAGCT gives the number of generations cataloged.
12 (C)	4	CTGWAGB	Address of a 4-byte field that contains the generation level to be used if CTGSUPLT and CTGGDGL in the CTGPL are set to one. If CTGGDGL is zero, the latest generation is used.

PCCB—Private Catalog Control Block

A PCCB describes each user's catalog to the OS/VS system. The PCCB is built when the user's catalog is opened. If the catalog is already open for another user, a PCCB is built when the user's JCL DD STEPCAT or JOBCAT statement specifies the catalog. The JSCB associated with the user task's TCB points to the first PCCB. Other PCCBs available to the user's task are chained from the first PCCB.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	4	PCCACRO	Acronym of the control block—C'PCCB'
4 (4)	4	PCCNEXTP	Address of the next PCCB in the PCCB chain, or zero.
8 (8)	4	PCCPREVP	Address of the previous PCCB in the PCCB chain, or zero.
12 (C)	4	PCCSTATS	Indicator flags
12 (C)	1	PCCSTAT1	Flag byte 1:
	1...	PCCSTEPC	The catalog represented by this PCCB is associated with a STEPCAT DD statement.
	.1..	PCCALIAS	This PCCB contains an alias name for a user's catalog. The catalog's DSNAME is also in the PCCB.
	..1.	PCCACTIV	The catalog represented by this PCCB is allocated and active.
	...1	PCOSCVOL	The catalog is an OS CVOL catalog
	...x xxxx		Reserved
13 (D)	3		Reserved
16 (10)	4	PCCACBP	Address of the ACB for the user's catalog.
20 (14)	8	PCCDDNAM	DDNAME for a dynamically allocated catalog.
28 (1C)	44	PCCDSNAM	Catalog's DSNAME
72 (48)	44	PCCTGCON	Catalog's alias (alternate DSNAME)
116 (74)	6	PCVOLSER	OS CVOL's volume serial number
122 (7A)	54		Reserved

PCTT—Private Catalog Termination Table

The PCTT records the open status of user catalogs, and it enables proper cleanup and freeing of control blocks during task termination. The PCTT is built dynamically by IDACAT11, mapped by IDAPCTT, and pointed to by ASCBPCTT. There is one PCTT for each address space.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	4	PCTTID	Acronym of the table—C'PCTT'
4 (4)	4	PCTTENBR	Number of slots for ACB addresses
8 (8)	4	PCTURR	Number of slots used for ACB addresses
12 (C)	4		Reserved
16 (10)	1	PCTFLGS1	Flag byte
	1.......	PCTTCL	Temporary close
	.xxxxxxx		Reserved
17 (11)	1		Reserved
18 (12)	2	PCTTOPCT	Open count
20 (14)	4	PCTTACB	ACB address slot

RPL—Request Parameter List

The RPL contains user-request information and error feedback information. It also maintains information required by GET and PUT macro instructions.

Offset	Bytes and Bit Pattern	Field Name	Description
0 (0)	4	RPLIDWD	Identification word of the RPL:
0 (0)	1	RPLID	RPL identifier, X'00'
1 (1)	1	RPLSTYP	RPL subtype, X'10'
2 (2)	1	RPLREQ	Request type:
	X'0'	RPLGET	GET request
	X'1'	RPLPUT	PUT request
	X'2'	RPLCHECK	CHECK request
	X'3'	RPLPOINT	POINT request
	X'4'	RPLENDRE	ENDREQ request
	X'5'	RPLERASE	ERASE request
	X'6'	RPLVERIF	VERIFY request
	X'8'	RPLPFMTD	Preformat data
	X'9'	RPLPFMTI	Preformat index
	X'A'	RPLFRCIO	Force I/O request
	X'B'	RPLGETIX	GETIX request
	X'C'	RPLPUTIX	PUTIX request
	X'D'	RPLSRCHB	Search buffer request
	X'E'	RPLMRKB	Mark buffer request
	X'F'	RPLWRTB	Write buffer request
3 (3)	1	RPLLEN	Length of the RPL
4 (4)	4	RPLPLHPT	Address of the PLH
8 (8)	4	RPLECB	Address of the external ECB, or an internal ECB:
	1...	RPLWAIT	The event has not yet completed
	.1..	RPLPOST	The event has completed
	..xx xxxx		Reserved
9 (9)	3		Reserved, if RPLECB is an internal ECB.
12 (C)	4	RPLFDBWD	Feedback word.

Request Parameter List (RPL)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
12 (C)	1	RPLSTAT	RPL status flags:
	.1..	RPLCHKI	CHECK has been issued
	..1.	RPLEDRQI	ENDREQ has been issued
	x..x xxxx		Reserved
13 (D)	3	RPLFDBK	RPL feedback area (See "Diagnostic Aids" in *OS/VS2 VSAM Logic* for a list of RPL return codes and condition codes.)
13 (D)	1	RPLRTNCD RPLERREG	RPL return code
14 (E)	2	RPLCNDCD	RPL condition code
14 (E)	1	RPLCMPON	Component issuing the code
15 (F)	1	RPLERRCD	Error code
16 (10)	2	RPLKEYLE RPLKEYL	Key length
18 (12)	2	RPLSTRID	RPL string identifier
20 (14)	4	RPLCCHAR	Address of the control character
24 (18)	4	RPLDACB	Address of the caller's ACB
28 (1C)	4	RPLTCBPT	Address of the user's TCB
32 (20)	4	RPLAREA	Address of the caller's record area
36 (24)	4	RPLARG	Address of the caller's search argument
40 (28)	4	RPLOPTCD	Option flags
40 (28)	1	RPLOPT1	Option flag byte 1:
	1...	RPLLOC	Locate mode
	0...		Move mode
	.1..	RPLDIR	Direct-search access
	..1.	RPLSEQ	Sequential access
	...1	RPLSKP	Skip sequential processing
	.... 1...	RPLASY	Asynchronous request
	.... 0...		Synchronous request
	.... .1..	RPLKGE	Search key greater than or equal
	.... .0..		Search key equal
	.... ..1.	RPLGEN	Generic key
	.... ..0.		Full key
	1	RPLECBSW	The RPLECB field contains the ECB's address
41 (29)	1	RPLOPT2	Option flag byte 2:
	1...	RPLKEY	Locate the record identified by a key
	.1..	RPLADR RPLADD	Locate the record at the caller-specified relative byte address (RBA)
	..1.	RPLCNV	Locate the control interval at the caller-specified RBA
	...1	RPLBWD	Process in backward direction
	...0		Process in forward direction
	.... 1...	RPLLRD	Locate or retrieve the last record in the data set
	.... 0...		Locate, retrieve, or store the record identified by the user's argument
	.... ..1.	RPLUPD	Update processing
	1	RPLNSP	Note the string position
	.... .x..		Reserved

Request Parameter List (RPL)—Description and Format (continued)

Offset	Bytes and Bit Pattern	Field Name	Description
42 (2A)	1	RPLOPT3	Option flag byte 3:
	1...	RPLEODS	End of the user's output data set
	.1..	RPLSFORM	Special form on remote printer
	..1.	RPLBLK	Block the records
	..0.		The records are unblocked
	...1	RPLVFY	UCS/FCB verify
	.... 1...	RPLFLD	UCS fold
	.... .xx.	RPLFMT	Format type:
	.... .00.		UCS load
	.... .01.		FCB load
	.... .10.		Reserved
	.... .11.		Reserved
	1	RPLALIGN	Align the buffer and notify the operator
	0		Do not align the FCB buffer loads
43 (2B)	1	RPLOPT4	Reserved
44 (2C)	4	RPLNXTRP RPLCHAIN	Address of the next RPL in the chain
48 (30)	4	RPLRLEN	Length of the record
52 (34)	4	RPLBUFL	Length of the user's buffer
56 (38)	4		Reserved
60 (3C)	8	RPLRBAR	RBA return location
60 (3C)	2	RPLAIXPC	Alternate index pointer count
62 (3E)	1	RPLAIXID	Alternate index pointer type:
	1...	RPLAXPKP	Relative byte address
	0...		Prime key pointer
	.xxx xxxx		Reserved
63 (3F)	1		Reserved
64 (40)	4	RPLDDDD	Relative byte address
68 (44)	1		Reserved
69 (45)	1	RPLACTIV	CHECK not issued
70 (46)	2	RPLEMLEN	Error message area length
72 (48)	4	RPLERMSA	Address of the error message area

DIAGNOSTIC AIDS

This chapter provides several aids that can be useful when you are trying to diagnose difficulties with OS/VS2 catalog management modules. These aids include:

- A description of the cross-reference information published on microfiche cards.
- A list of messages issued by OS/VS2 catalog management, cross-referenced to enable you to detect the module causing the message to be issued.
- A list of macro instructions issued by catalog management and their functions.
- A catalog debug aid that provides options you can select and activate upon termination of a catalog management request.
- A description of a system-provided service (GTF—the generalized trace facility).
- A description of the catalog communication area's register save area.
- A list of catalog management return codes and error codes.

Additional aids can be found in other parts of the book and in the program listings. These include:

- Register contents on entry to a module, which are under "INPUT" in the module's prologues.
- Use of registers and equated names for registers, which can be found under "NOTES" in the module's prologues.
- Error codes, which are under "EXIT-ERROR" in the module's prologues.
- A list of modules, their external procedure names, and their associated method of operation diagrams and program organization figures, which is in the "Module Directory."
- A list of external procedure names and their modules, which is in the "External Procedure Name Directory."
- A list of all the information in this book that is related to a module, which is in the "Index."
- A definition of terms and acronyms used in this book, and in the OS/VS2 catalog management listings, which is in the "Glossary."

Microfiche Cross-Reference Aids

Microfiche cards in *OS/VS2 Catalog Management Cross-Reference* contain valuable cross-reference information that you should be aware of. Two types of information are available:

- Symbolic-name usage table: lists each symbolic name that appears in the catalog management source-code listings, which lists each module that refers to the symbolic name, and specifies how each module refers to the symbolic name.
- Macro-instruction usage table: lists each macro-instruction that is issued in catalog management listings, specifies the total number of times the macro-instruction is issued, lists each module that issues the macro-instruction, and specifies the number of times the module issues the macro-instruction.

How To Read the Symbolic-Name Usage Table

Microfiche cards in the *OS/VS2 Catalog Management Cross-Reference* microfiche package contain the symbolic-name usage table, or Symbol Where Used Report, for OS/VS2 catalog management listings. Three kinds of information are available from the table, as shown in Figure 45:

- A list of symbolic names—this includes field names, symbolic address names, return code names, constant/value names, flag-bit names, etc—in alphanumeric order from top to bottom on the page.

  Note: In the lower-right corner of each page, the lowest and highest name for the page is shown.

DATE: 09/07/75 SYMBOL WHERE USED REPORT --- OS/VS2 VSAM CAT MGMT (EXTERNAL SYMBOLS) PAGE 3

SYMBOL	MODULE	ACCESS	MODULE	ACCESS	MODULE	ACCESS	MODULE	ACCESS	MODULE	ACCESS	MODULE	ACCESS
ACBBUFSP	IEAVNP1A	R										
ACBCAT	IDACAT11	W	IDACAT12	W	IGG0CLAD	W	IGG0CLAE	W				
ACBCBMWA	IDA019C1	DRW										
ACBCRNCK	IGC0A05B	C										
ACBCRNRE	IGC0B05B	C	IGC0C06C	C								
ACBDDNM	IDACAT11	W	IDA019C1	W	IEAVNP1B	R	IEAVNP11	W	IFG0191X	R	IGC0A05B	W
	IGG0CLAD	W	IGG0CLAE	W	IGG0CLBG	W						
ACBDEB	IDACAT13	R	IEAVNP1A	W	IGC0B05B	C						
ACBDIR	IEAVNP1A	C	IFG0191X	W								
ACBDORGA	IDACAT11	W	IDA019C1	W	IGC0C06C	C	IGG0CLAD	W	IGG0CLAE	W	IGG0CLBG	W
ACBDOSID	IGG0CLAD	W	IGG0CLAE	W	IGG0CLBG	W						
ACBDTFID	IGG0CLAD	R	IGG0CLAE	R	IGG0CLBG	R						
ACBERFL	IFG0191X	W										
ACBERFLG	IFG0191Y	W	IFG0200N	W	IGC0A05B	RWC	IGG0CLAE	C				
ACBEXFG	IDACAT11	W	IDA019C1	W	IGG0CLAD	W	IGG0CLAE	W	IGG0CLBG	W		
ACBEXLST	IGC0C06C	RW										
ACBID	IDACAT11	W	IDA019C1	WC	IGC0C06C	C	IGG0CLAD	W	IGG0CLAE	W	IGG0CLBG	W
ACBIDVAL	IGG0CLAD	R	IGG0CLAE	R	IGG0CLBG	R						
ACBIN	IDA019C1	W	IFG0191X	W	IGG0CLBG	W						
ACBINFL	IEAVNP1A	R	IFG0191X	W								
ACBINRTN	IGC0B05B	R										
ACBKEY	IDA019C1	W	IFG0191X	W	IGG0CLBG	W						
ACBLEN	IDACB2	M										
ACBLENG	IDACAT11	W	IDA019C1	W	IGG0CLAD	W	IGG0CLAE	W	IGG0CLBG	W		

ACCESS CODES: D = DEFINITION, R = READ, W = WRITE, C = COMPARE, E = EQUATE OPERAND, M = MACRO, A = ABSOLUTE, P = PARAMETER

SYMBOL ENTRIES: ACBBUFSP - ACBLENG

Figure 45. Symbolic Name Usage Table

- A list of modules that refer to each symbolic name, in alphanumeric order from left to right across the page.
- A code indicating how each module refers to the symbolic name:

W–WRITE	The data field or bit value was modified by at least one line of code in this module. If the module contains a statement: A = B then the module's use of 'A' is to modify it ('A' appears to the left of an equate sign in a statement that is not an 'IF' statement.)
R–READ	The data field or value was referred to by at least one line of code in this module. If the module contains a statement: A = B then the module's use of 'B' is to refer to it, using it to modify 'A' ('B' appears to the right of an equate sign in any type of statement).
C–COMPARE	The data field or value was compared against another value. If the module contains a statement: IF A = B, THEN ... then the module's use of 'A' is to compare it to 'B' ('A' appears to the left of an equate sign in an 'IF' statement). Note that the module's use of 'B' is to refer to it, not to compare it.

Other codes are explained in the "Access Codes" at the bottom of each page in the table.

How To Read the Macro-Instruction Usage Table

A microfiche card in the *OS/VS2 Catalog Management Cross-Reference* microfiche package contains the macro-instruction usage table, or Macro Where Used Report, for OS/VS2 catalog management listings. Three kinds of information are available, as shown in Figure 46:

- A list of macro-instruction names in alphanumeric order from top to bottom.
- A list of the modules that issue each macro-instruction, in alphanumeric order from left to right across the page.
- The total number of times all catalog management modules issued the macro-instruction, and the number of times each module in the list issued the macro-instruction.

DATE: 09/07/75 MACRO WHERE USED REPORT --- OS/VS2 VSAM CAT MGMT PAGE 1

MACRO	TOTAL #	MODULE	#	MODULE	#	MODULE	#	MODULE	#	MODULE	#	MODULE	#
ABEND	1	IEAVNP11	1										
ABPCALL	1	IEAVNP12	1										
ACB	1	IGC0A05B	1										
ACBX	1	IEAVNP11	1										
ADDRESS	1	IGG0CLAP	1										
AMCBS		IDACAT11	1	IDACAT12	1	IDACAT13	1	IEAVNP1A	1	IEAVNP1B	1	IEAVNP11	1
	8	IEAVNP12	1	IGGMCDCL	1								
AMOREGN	1	IGGMCDCL	1										
ASM1	2	IEAVNP1A	1	IEAVNP11	1								
BLDL	2	IEAVNP11	2										
CALL	5	IDACB2	1	IGG0CLAQ	4								
CARD	1	IGG0CLAP	1										
CATGODSP	1	IGG0CLAP	1										
CATG0GC1	1	IGG0CLAP	1										
CATGOSEQ	1	IGG0CLAP	1										
CCAASCIK	1	IGG0CLAP	1										
CCACD1	2	IGG0CLAQ	2										
CCACPE2	1	IGG0CLAP	1										
CCACPE3	1	IGG0CLAP	1										
CCACPE4	1	IGG0CLAP	1										
CCACPE5	1	IGG0CLAP	1										
CCACPE6	2	IGG0CLAP	2										
CCACPE7	1	IGG0CLAP	1										
CCARABB	1	IGG0CLAP	1										
CCARABFL	1	IGG0CLAP	1										

MACRO ENTRIES: ABEND - CCARABFL

Figure 46. Macro-Instruction Usage Table

Messages

Messages IDA001 through IDA022 are macro-instruction messages and refer to an incorrectly coded macro instruction.

Message Number	Message Text
IDA001	INVALID POSITIONAL PARAMETER, *xxx* - IGNORED
IDA002	*xxx* KEYWORD REQUIRED - NOT SPECIFIED
IDA003	INVALID VALUE, *yyy*, SPECIFIED FOR *xxx* KEYWORD
IDA004	*xxx* KEYWORD NOT VALID FOR EXECUTE FORM - IGNORED
IDA005	INVALID SUBLIST ITEM FOR *xxx* KEYWORD, *yyy*
IDA006	*xxx* VALUE, *yyy*, NOT VALID FOR LIST FORM
IDA007	LOGIC ERROR IN MACRO *xxx*
IDA008	INCOMPATIBLE SUBLIST ITEMS, *yyy* AND *zzz* FOR *xxx* KEYWORD
IDA009	*xxx* CONTROL BLOCK KEYWORDS SPECIFIED - ONLY ONE ALLOWED
IDA010	EXIT ADDRESS REQUIRED FOR *xxx* KEYWORD - NOT SPECIFIED
IDA011	*xxx* IS NOT A VALID *yyy* KEYWORD - IGNORED
IDA018	VTAM KEYWORD, *xxx*, SPECIFIED WITHOUT SPECIFYING AM=VTAM
IDA019	KEYWORDS *xxx* and *yyy* ARE INCOMPATIBLE

Messages (continued)

Message Number	Message Text
IDA020	VTAM SUBLIST ITEM, *xxx*, SPECIFIED FOR *yyy* KEYWORD WITHOUT SPECIFYING AM=VTAM
IDA021	*xxx* and *yyy* KEYWORDS MUST BE SPECIFIED TOGETHER, BUT ONE IS MISSING
IDA022	CONFLICTING SUBLIST ITEMS WERE SPECIFIED FOR *xxx* KEYWORD

Message Number	Message Text	Detected by	Issued by
IEC101A	M *ddd,ser,jjj,sss,dsn*	IGG0CLBL	IDA0192V
IEC111E	D, *ddd,ser*	IGG0CLBL	IDA0192V
IEC113A	ENTER PASSWORD FOR DATA SET *dsn*	IGG0CLBG IGG0CLBM IGG0CLB6	
IEC114E	D *ddd* [,*ddn-n*]	IGG0CLBL	
IEC115I	INVALID PASSWORD	IGG0CLBG IGG0CLBM IGG0CLB6	
IEC116A	REENTER	IGG0CLBG IGG0CLBM IGG0CLB6	
IEC130I	*ddn* - DD STATEMENT MISSING	IFG0191X	
IEC301A	S JOB *xxxxxxxx* DSNAME *dsn*	IGG0CLBM	
IEC331I	*ccc-000,jjj,sss,ffff*	IGG0CLAG (IGGPIORA)	IGG0CLAF (IGGPEMSG)
IEC332I	*fff* [,*ffff*...]	IGG0CLAG (IGGPIORA)	IGG0CLAF (IGGPEMSG)
IEC333I	*teee,xx,ddd,iii*	IGG0CLAG (IGGPIORA)	IGG0CLAF (IGGPEMIO)

Macro Instructions

The following tables list VSAM and OS/VS macro instructions and explain what they do. The macro instructions are divided into those that define control blocks and data areas (mapping macro instructions) and those that result in executable code (action macro instructions).

Mapping Macro Instructions

The following table lists macro instructions that define the format of control blocks and data areas used by VSAM modules.

Macro Instruction	Description and Issuing Modules
AMCBS	Maps the OS/VS2 catalog vectors table (AMCBS) See "Data Areas" for a description of the AMCBS.
CVT	Maps the communications vector table (CVT)
F4DSCB	Maps the format-4 Data Set Control Block (DSCB)
IDAAMB	Maps the Access Method Block (AMB) See "Data Areas," in *OS/VS2 VSAM Logic*, for a description of the AMB.

Macro Instructions That Define Data Areas (continued)

Macro Instruction	Description and Issuing Modules
IDAAMBL	Maps the Access Method Block List (AMBL) See "Data Areas," in *OS/VS2 VSAM Logic*, for a description of the AMBL.
IDAAMDSB	Maps the Access Method Data Set Statistics Control Block (AMDSB) See "Data Areas," for a description of the AMDSB.
IDAARDB	Maps the Address Range Definition Block (ARDB) See "Data Areas," in *OS/VS2 VSAM Logic*, for a description of the ARDB.
IDABUFC	Maps the buffer control block (BUFC) See "Data Areas," in *OS/VS2 VSAM Logic*, for a description of the BUFC.
IDADIWA	Maps the data insert work area (DIWA) See "Data Areas," in *OS/VS2 VSAM Logic*, for a description of the DIWA.
IDAFOREC	Maps the work area for VSAM Open/Close/EOV Modules IDAFOREC issues IDAPDPRM, IEFJFCBN, and IEFJFCBX.
IDAIOMB	Maps the I/O management control block See *OS/VS2 Data Areas* for a description of the IOMB.
IDAPCTT	Maps the private catalog termination table (PCTT) See "Data Areas" for a description of the PCTT.
IDAPDPRM	Maps the VSAM Open/Close/EOV problem determination parameter list
IDAPLH	Maps the placeholder (PLH) See "Data Areas," in *OS/VS2 VSAM Logic*, for a description of the PLH.
IDARMRCD	Lists the record management return codes
IDAVVOLL	Maps the VSAM Open/EOV: Volume Mount and Verify volume serial number list
IECDIOSB	Maps the I/O supervisor control block See *OS/VS2 Data Areas* for a description of the IOSB.
IECSDSL1	Maps the SDSL1
IEESMCA	Maps the SMCA
IEFJESCT	Maps the JESCT
IEFJFCBN	Maps the Job File Control Block (JFCB)
IEFJFCBX	Maps the Job File Control Block (JFCB)
IEFJMR	Maps the JMR
IEFPCCB	Maps the Private Catalog Control Block (PCCB) See "Data Areas" for a description of the PCCB.
IEFQMIOP	Maps the QMIOP
IEFTCT	Maps the TCT
IEFTIOT1	Maps the task input/output table (TIOT)
IEFUCBOB	Maps the OS/VS Unit Control Block (UCB)
IEZCTGCV	Maps the catalog control volume list (CTGCV) See "Data Areas" for a description of the CTGCV.
IEZCTGFL	Maps the catalog field parameter list (CTGFL) See "Data Areas" for a description of the CTGFL.
IEZCTGFV	Maps the catalog field vector table (CTGFV)

Macro Instructions That Define Data Areas (continued)

Macro Instruction	Description and Issuing Modules
	See "Data Areas" for a description of the CTGFV.
IEZCTGPL	Maps the catalog parameter list (CTGPL) See "Data Areas" for a description of the CTGPL.
IEZCTGVL	Maps the catalog volume list (CTGVL) See "Data Areas" for a description of the CTGVL.
IEZCTGWA	Maps the catalog scheduler work area (CTGWA) See "Data Areas" for a description of the CTGWA.
IEZDEB	Maps the OS/VS Data Extent Block (DEB)
IEZIOB	Maps the OS/VS Input/Output Block (IOB)
IEZJSCB	Maps the OS/VS Job Step Control Block (JSCB)
IFGACB	Maps the access method control block (ACB) See "Data Areas" for a description of the ACB.
IFGRPL	Maps the request parameter list (RPL) See "Data Areas" for a description of the RPL.
IGGCAXWA	Maps the catalog auxiliary work area (CAXWA) See "Data Areas" for a description of the CAXWA.
IGGCCA	Maps the catalog communications area (CCA) See "Data Areas" for a description of the CCA.
IGGMCDCL	Contains the commonly used control block formats and constants for catalog management modules IGGMCDCL issues AMCBS, AM0REGN, COMREGN, CVT, IEZCTGCV, IEZCTGFL, IEZCTGFV, IEZCTGPL, IEZCTGWA, IFGACB, IGGCAXWA, IGGCCA, IGGMCTRC, and IKJTCB.
IGGMCMDM	Maps the catalog management commonly-used record structures
IGGMCMWA	Maps the catalog management services work area
IGGMCTRC	Lists the catalog management return codes
IGGMDRWA	Maps the catalog DSCB read-in work area
IGGMF4WA	Maps the Catalog Management format-4 DSCB work area
IGGMGVO	Maps the volume information set of fields
IGGMSAWA	Maps the Catalog Management: Suballocate work area
IGGMUPDE	Defines the commonly-used declarations for VSAM Catalog Management: Update-Extend modules IGGMUPDE issues IDAAMDSB, IGGMCDCL, IGGMCMDM, IGGMSAWA, and IGGMVEDC.
IGGMVEDC	Maps the Volume Catalog Record
IGGSHWPL	Maps the SHOWCAT input/output parameter list.
IHADCB	Maps the OS/VS Data Set Control Block (DCB)
IHADCBDF	Maps the OS/VS Data Set Control Block (DCB)
IHARB	Maps the OS/VS Request Block (RB)
IHASRB	Maps the VS2 service request block See *OS/VS2 Data Areas* for a description of the SRB.
IKJPSCB	Maps the OS/VS PSCB
IKJTCB	Maps the task control block (TCB)
SGIDA401	Lists the VSAM SYSGEN global definitions
XCTLTABL	Maps the OS/VS XCTL table

Action Macro Instructions

This table lists the macro instructions that generate executable code.

Macro Instruction	Description
ADDREC	Calls IGGPPAD to write a new record into the catalog
CALLSF(xxxx)	Transfers control to procedure IGGPxxxx
CALL EXIT	Returns control to the caller of the procedure
CATLG	Loads the address of the catalog parameter list (CTGPL) into register 1 and issues SVC 26
CATPROB	Problem determination
CLOSE	VSAM CLOSE: Disconnect a user from a VSAM data set
COMB	Generates combination name entries in the VSAM combination name index table
DELETE	(Same as OS/VS DELETE macro instruction)
DELREC	Calls IGGPPDE to erase a catalog record
DEQ	(Same as OS/VS DEQ macro instruction)
DEVTYPE	Determine the direct-access device type
DICT	Generates entries in the catalog field name dictionary
ENDREQ	Terminate a VSAM record processing request (such as GET or PUT)
ENQ	(Same as OS/VS ENQ macro instruction)
ERASE	Delete a VSAM record
EXCP	(Same as OS/VS EXCP macro instruction)
FREEMAIN	Releases virtual storage obtained by a GETMAIN
GET	Retrieves a record from a data set on a direct-access device
GETMAIN	Obtains virtual storage for a temporary work area
GETREC	Calls IGGPGET to retrieve a catalog record
IGGMEND	Generates code at the end of Catalog Management modules
IGGMODUL	Generates header code for Catalog Management modules
IGGMPROC	Generates header code for Catalog Management procedures
LOAD	(Same as OS/VS LOAD macro instruction)
MODESET	(Same as OS/VS MODESET macro instruction)
OPEN	Connect a user's program to a VSAM data set
PUT	Write a record into the VSAM data set
RESERVE	(Same as OS/VS RESERVE macro instruction)
SMFWTM	Write the SMF message into the SMF data set
SYNCH	(Same as ISAM SYNCH macro instruction)
TIME	Obtain the correct time from the OS/VS system time-of-day clock
WAIT	(Same as OS/VS WAIT macro instruction)
WTO	Write a message to the operator (no reply)
WTOR	Write a message to the operator (a reply is expected)
XCTL	Transfer control (OS/VS XCTL macro instruction)

Using the VSAM Catalog Debug Aid

The VSAM debug catalog aid allows the PSR to exercise certain options when VSAM catalog management requests terminate. These options include trapping, issuing a problem determination message, and retaining the CCA across ABEND dumps. They can be selected in any combination, and you can specify that the trap and message options be activated upon termination of (1) all requests, (2) only those requests that generate a nonzero return code in CCACD1, (3) only those requests that generate an abnormal return code in CCACD1, or (4) only those requests that generate a specific return code in CCACD1.

Defining Debug Aid Options

Debug aid options are defined by storing values and setting bits within the CVTAMFF field in the CVT. The PSR can use the CPU manual procedure AM (alter main storage) to modify the CVTAMFF field. (Note, however, that bits 0-3 of CVTAMFF must not be changed.)

You accomplish debug activity by storing a nonzero value (X'01' – X'FF') into CVTAMFF + 1 (the CVT's location + X'109') and X'07FE' (a BR 14 instruction) into bytes 3 and 4 of CVTAMFF (CVT's location + X'10A'). The nonzero value you store determines the scope of the debug activity, as follows:

X'01'	Exercise options upon termination of all requests.
X'02'	Exercise options only when the catalog return code is nonzero.
X'03'	Exercise options only when the catalog return code is not a normal return code (0, 8, 36, 40, 44, 76, 140 (reason code 40), 188, and 240 are considered normal).
X'04' – X'FF'	Exercise options only when the catalog return code equals the value stored.

Selecting Debug Options

Each option selected will be exercised only when the catalog termination routine determines that the catalog return code (in CCACD1) falls within the defined scope.

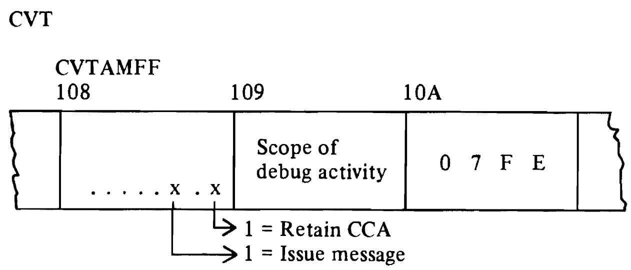
The trap option is activated by setting a hardware address stop, a DSS "AT", or a VM ADSTOP at the location of the BR 14 instruction contained in the low-order two bytes of CVTAMFF (the CVT's location + X'10A'). The catalog termination routine executes a BALR R14, R15 instruction to pass control to the BR 14 instruction.

Register contents at the time of the debug trap are:

Register	Contents
0	CPL bytes 0, 1, 2, and 16 (the type of catalog management request can be derived from the information contained in these bytes)
1	Contents of CCAPROB (module ID, error code, and return code)
11	Pointer to CCA
14	Return address
15	Address of trap instruction

To cause determination message IEC331I to be issued, set bit 5 (X'04') of CVTAMFF (the CVT's location + X'108') to 1.

To include the CCA in ABEND dumps, set bit 7 (X'01') of CVTAMFF to 1. This option can be used independently of the other options.



Generalized Trace Facility

The Generalized Trace Facility (GTF) can be used to record information about VSAM processing at the time of an error. If GTF is active in the OS/VS system, GTF is used to trace VSAM control blocks when there is an error.

GTF is used to record the contents of the ACB, AMBL, AMBs, AMDSBs, and TIOT entry for the data set being processed when the error occurred.

To format and print GTF records, use the PRDMP service aid with "USR=(FFF,FF5)" specified in the EDIT statement.

Additional information on GTF and PRDMP is contained in *OS/VS2 System Programming Library: Service Aids*.

Catalog Communication Area Register Save Area

A catalog communication area (CCA) is built for every call to catalog management. The CCA contains a register save area (CCAREGS) that allows the PSR (programming systems representative) to follow the flow of control from one catalog management procedure to another, through each procedure called to process the request.

The contents of registers 12, 13, and 14 are put into CCAREGS whenever a catalog management procedure is entered. The current value of register 13 is the address of the latest entry in CCAREGS.

If an external catalog management procedure is entered from another catalog management procedure, three words are saved as follows:

- the first word contains the contents of register 12—the calling procedure's base address,
- the second word contains the contents of register 13—a pointer to the previous 12-byte entry in the register save area (CCAREGS), and
- the third word contains the contents of register 14—the return address in the calling procedure.

Immediately after registers 12, 13, and 14 are saved (at register 13 + 12 (decimal)), register 12 is updated to contain the "called" procedure's base address. Register 13's value is increased by 12, so that it points to the latest entry in CCAREGS. While a catalog management procedure is processing, register 11 contains a pointer to the beginning of the CCA.

Note that backward movement is not recorded in the trace table. For example, if procedure B returns to procedure A, the return is not shown in the register save area.

Error Return Codes

VSAM sets error codes in the RPL and the ACB. Error codes set in the RPL are listed and explained in *OS/VS Virtual Storage Access Method (VSAM) Programmer's Guide* and in *OS/VS2 VSAM Logic.* Those set in the ACB to indicate open, close, or end of volume errors are listed and explained in *OS/VS2 VSAM Logic.*

OS/VS2 catalog management sets error and reason codes in CCAPROB. (For a description of the CCA, see "Catalog Management Control Block Descriptions" in the "Data Areas" section of this publication.) CCAPROB includes an identification of the catalog management module that set the code (CCAMODID), a reason code (CCAREASN), and a return code (CCACD1), which appears in register 15. Complete explanations of the error and return codes, together with the appropriate programmer responses are given in the description of message IDC3009I in *OS/VS Message Library: VS2 System Messages*. Brief descriptions of the return codes are given below:

Catalog Management Error Codes

Return Code Set in CCA's CCAPROB Field	Symbolic Name	Meaning of Return Code
0 (0)	RCS	Operation completed successfully.
4 (4)	RCCAT	An error occurred while performing open/close processing for a VSAM catalog or catalog recovery area. If no STEPCAT or JOBCAT DD statement was supplied, dynamic allocation may have failed because the volume is allocated exclusively. The volume may be allocated exclusively because (1) the DD statement specified deferred mount, (2) the volume count is greater than the unit count on the DD statement, or (3) a previous Access Method Services command demounted the volume and changed the allocation of the volume to exclusive.
8 (8)	RCENT	Entry does not exist, if action is one that locates an entry, or entry already exists, if action is one that adds an entry to a catalog.
20 (14)	RCINSP	Not enough space is available in the catalog data set. Another extent cannot be obtained, because there is no more space on the volume in which the catalog resides or the maximum number of extents has been reached.
24 (18)	RCIOL	Permanent read error in VSAM catalog.
28 (1C)	RCIONL	Permanent I/O error in VSAM catalog.
32 (20)	RCINCPL	Error was detected in the catalog parameter list (CTGPL).
36 (24)	RCDSNF	Data set was not found.
40 (28)	RCVLSZ	Volume list or work area is too small—the required length value is returned in the feedback field.

Catalog Management Error Codes

Return Code Set in CCA's CCAPROB Field	Symbolic Name	Meaning of Return Code
44 (2C)	RCVLSM	Work area is too small; system is unable to return required size.
48 (30)	RCINFUNC	Operation is not a valid one.
52 (34)	RCIOU	I/O error was detected on a user volume. An attempt to modify the VTOC of the volume on which a user-specified data set is being defined or modified failed because of a read or write error.
56 (38)	RCSEC	Password is incorrect.
60 (3C)	RCINENT	Catalog record type is invalid.
64 (40)	RCNAME	Data set or index catalog record associated with the cluster or alternate index catalog record was not found.
68 (44)	RCNOSP	No space is available on a user volume.
72 (48)	RCNMNTD	User volume is not mounted.
76 (4C)	RCNUNIT	Unit is not available for mounting user volume or volume not mounted.
80 (50)	RCRELOP	Invalid related object. The object specified in the RELATE parameter of the DEFINE command does not exist or is improper for the object being defined.
84 (54)	RCDATE	Purge date has not expired.
88 (58)	RCCRAOP	Error with a catalog recovery area define operation.
92 (5C)	RCDSEXT	Data set has reached the maximum number of extents.
94 (5E)	RCOBTAIN	An OS/VS DADSM Obtain request failed during a VSAM catalog delete request.
96 (60)	RCSPANCK	Error in specifying key length, key position, or record size for an alternate index or a spanned cluster.
98 (62)	RCRENAME	An unusual condition occurred during ALTER name of a unique or nonVSAM data set.
102 (66)	RCSCRTCH	An OS/VS DADSM Scratch request failed during a VSAM delete request for a unique or nonVSAM data set.
104 (68)	RCCATEX	Catalog exists.
106 (6A)	RCNTFMT4	A format-4 DSCB processing error was encountered.
108 (6C)	RCINFNAM	Field name is invalid.
112 (70)	RCINFPL	Field parameter list (CTGFL) contains invalid parameters.
116 (74)	RCCATBAL	Catalog records are invalid.
120 (78)	RCSYSFLD	User attempted to modify a system field or nonexistent field.
124 (7C)	RCINCI	Control interval number is invalid.
128 (80)	RCBLKVCK	User provided a work area outside his address space.
132 (84)	RCINPTR	Pointer is not valid.
136 (88)	RCMISPAR	Required parameter is missing.
140 (8C)	RCINCNPM	Specified parameters are inconsistent or conflicting.
144 (90)	RCINENTN	Entry name is invalid.

Catalog Management Error Codes

Return Code Set in CCA's CCAPROB Field	Symbolic Name	Meaning of Return Code
148 (94)	RCVOLOWN	Volume is already owned by another VSAM catalog.
152 (98)	RCDNECAT	User attempted to delete a catalog that is not empty.
156 (9C)	RCNOSPSA	No space available to suballocate.
160 (A0)	RCVNDSPD	Deletion of space object did not cause volume to be deleted.
164 (A4)	RCINSSWA	Not enough storage is available for work area.
168 (A8)	RCINVDTY	Specified device-type is not supported.
172 (AC)	RCDUPNVL	Volume has duplicate data space name.
176 (B0)	RCNSPVTC	No space available on VTOC for DSCB.
180 (B4)	RCDSNFND	Data space was not found.
184 (B8)	RCDSO	Data set is currently open, so the catalog record cannot be modified.
188 (BC)	RCCATUNA	The catalog is unavailable.
192 (C0)	RCMLRSZ	Maximum logical record length specified is greater than 32,761 for a nonspanned data set.
196 (C4)	RCMCISZD	Data component control interval size specified is greater than 32,767.
200 (C8)	RCMCISZI	Index component control interval size specified is greater than maximum block size of index device.
204 (CC)	RCKEYINC	Key extends beyond end of record.
208 (D0)	RCBUFSIZ	Buffer size is too small.
212 (D4)	RCSIZCAL	Control interval size cannot be calculated.
216 (D8)	RCVTCBAL	Volume's VTOC is invalid.
220 (DC)	RCDOSVTC	DOS VTOC cannot be converted to OS/VS VTOC.
224 (E0)	RCMXGRP	Catalog record has exceeded the maximum number of sets of fields allowed.
226 (E2)	RCTSAUTH	Test authorization failed.
228 (E4)	RCLOCKER	Error detected in time-of-day clock.
230 (E6)	RCHIGH	VSAM catalog retrieve of a control interval failed to get a low-range record from the VSAM catalog.
232 (E8)	RCSMFER	Error detected in SMF processing.
234 (EA)	RCLEOD	End of data encountered while reading the low data key range of the VSAM catalog.
236 (EC)	RCSPMAPE	Error detected in scanning the space map.
238 (EE)	RCNOUCEN	No user catalog entry in the master catalog for convert volume processing.
240 (F0)	RCINDER	Required DD statement missing.
242 (F2)	RCEFRMPH	A physical I/O error occurred during an erase of the data set being deleted.
244 (F4)	RCEF	Erase operation failed—DELETE operation was not performed.

Catalog Management Error Codes

Return Code Set in CCA's CCAPROB Field	Symbolic Name	Meaning of Return Code
248 (F8)	RCVOLENT	The volume catalog record (identified with a caller-specified volume serial number) was not found.
250 (FA)	RCEFRM	VSAM record management found a logical error during erase processing while deleting a VSAM data set.
252 (FC)	RCEE	Error was detected, and the operation was not completed.

Alphabetic List of the Catalog Management Error Return Code Symbolic Names

Name	Code	Name	Code
RCBLKVCK	128 (80)	RCIOU	52 (34)
RCBUFSIZ	208 (D0)	RCKEYINC	204 (CC)
RCCAT	4 (04)	RCLEOD	234 (EA)
RCCATBAL	116 (74)	RCLOCKER	228 (E4)
RCCATEX	104 (68)	RCMCISZD	196 (C4)
RCCATUNA	188 (BC)	RCMCISZI	200 (C8)
RCCRAOP	88 (58)	RCMISPAR	136 (88)
RCDATE	84 (54)	RCMLRSZ	192 (C0)
RCDNECAT	152 (98)	RCMXGRP	224 (E0)
RCDOSVTC	220 (DC)	RCNAME	64 (40)
RCDSEXT	92 (5C)	RCNMNTD	72 (48)
RCDSNF	36 (24)	RCNOSP	68 (44)
RCDSNFND	180 (B4)	RCNOSPSA	156 (9C)
RCDSO	184 (B8)	RCNOUCEN	238 (EE)
RCDUPNVL	172 (AC)	RCNSPVTC	176 (B0)
RCEE	252 (FC)	RCNTFMT4	106 (6A)
RCEF	244 (F4)	RCNUNIT	76 (4C)
RCEFRM	250 (FA)	RCOBTAIN	94 (5E)
RCEFRMPH	242 (F2)	RCRELOP	80 (50)
RCENT	8 (08)	RCRENAME	98 (62)
RCHIGH	230 (E6)	RCS	0 (0)
RCINCI	124 (7C)	RCSCRTCH	102 (66)
RCINCNPM	140 (8C)	RCSEC	56 (38)
RCINCPL	32 (20)	RCSIZCAL	212 (D4)
RCINDER	240 (F0)	RCSMFER	232 (E8)
RCINENT	60 (3C)	RCSPANCK	96 (60)
RCINENTN	144 (90)	RCSPMAPE	236 (EC)
RCINFNAM	108 (6C)	RCSYSFLD	120 (78)
RCINFPL	112 (70)	RCTSAUTH	226 (E2)
RCINFUNC	48 (30)	RCVLSM	44 (2C)
RCINPTR	132 (84)	RCVLSZ	40 (28)
RCINSP	20 (14)	RCVNDSPD	160 (A0)
RCINSSWA	164 (A4)	RCVOLENT	248 (F8)
RCINVDTY	168 (A8)	RCVOLOWN	148 (94)
RCIOL	24 (18)	RCVTCBAL	216 (D8)
RCIONL	28 (1C)		

GLOSSARY

Following is a list of acronyms and abbreviations of terms used in this book and in the VSAM code listings. If you do not find the term you are looking for, refer to the index or to the publication, *IBM Data Processing Glossary*, GC20-1699.

Acronyms and Abbreviations

ABEND: abnormal end

ACB: access method control block

ADR: addressed processing or addressed

ADDR: same as ADR

AIX: alternate index

AMB: access method block

AMBL: access method block list

AMCBS: access method control block structure control block

AMDSB: access method data statistics block

ARDB: address range definition block

ASCB: address space control block

ASM: auxiliary storage manager

ATIOT: alternate task input/output table

BISAM: basic indexed sequential access method

BLK: block, control interval

BUFC: buffer control block

CA: control area

CAXWA: catalog auxiliary work area

CCA: catalog communication area

CCB: command control block

CI: control interval

CIDF: control interval definition field

CMS: VSAM catalog management services

CNV: control interval or control-interval processing

CPA: channel program area

CPL: catalog parameter list (CTGPL)

CRA: catalog recovery area

CTGCV: catalog control volume list

CTGFL: field parameter list (FPL)

CTGFV: field vector table (FVT)

CTGPL: catalog parameter list (CPL)

CTGVL: catalog volume list

CTGWA: catalog work area

CVT: communications vector table

DCB: data control block

DDNAME: data definition name

DEB: data extent block

DIR: direct processing

DIWA: data insert work area

DSCB: data set control block

DSNAME: data set name

DSORG: data set organization

ECB: event control block

EDB: extent definition block

ENDREQ: end-the-request command

EOD: end of data

EOF: end of file

EOV: end of volume

ERFLG: error flags

ESDS: entry-sequenced data set

EXCD: exceptional conditions

EXCP: execute channel program

EXLST: exit list

Ext Proc: external procedure

FKS: full key search

Fn: format n

FPL: field parameter list (CTGFL)

FS: free space

FVT: field vector table (CTGFV)

GC: type code (group code)

GDG: generation data group

GEN: generic key search

GET: retrieve-a-record command

GO: Set of fields (group occurrence)

GOP: Set-of-fields pointer (group occurrence pointer)

ICWA: index create work area

ID: identifier

IICB: ISAM interface control block

IMWA: index modification work area

Int Proc: internal procedure

I/O: input/output

IOB: input/output block

ISAM: indexed sequential access method

IXSPL: index search parameter list

JFCB: job file control block

JSCB: job step control block

KEQ: search on key equal

KEY: keyed accessing

KGE: search on key greater or equal

KSDS: key-sequenced data set

L: link

LOC: locate

LPMB: logical-to-physical mapping block

MACR: macro instruction reference

MOD: module

MSS: Mass Storage System

n: integer number

NSI: next sequential instruction

NSP: next string position

NUP: no update

O/C/EOV: open/close/end of volume

OFLG: open flags

OPTCD: option code

OS/VS: operating system/virtual storage

PCCB: private catalog control block

PCTT: private catalog termination table

PL/I: programming language/one

PL/S: programming language/systems

PLH: placeholder list

PLHDR: placeholder header

PROC: procedure

PSR: programming systems representative

PSW: program status word

PUT: write-a-record command

QISAM: queued indexed sequential access method

RAB: record area block

RACF: resource access control facility

RB: request block

RBA: relative byte address

Rn: general-purpose register n

RDF: record definition field

REP: replication

RPL: request parameter list

RPLE: RPL extension for ISAM Interface processing

RRDS: relative record data set

RTN: routine

SCIB: search compressed index block

SEQ: sequential or sequential processing

SKP: skip sequential or skip sequential processing

SMF: system management facilities

STRNO: number of RPL strings

SVC: supervisor call

TCB: task control block

TIOT: task I/O table

TSO: time sharing option

UCB: unit control block

UPD: update mode (or data modify)

USVR: user security verification routine

VL: variable length

VM: virtual memory

VSAM: virtual storage access method

VSRT: VSAM shared resource table

VTOC: volume table of contents

WTG: where-to-go table

XCTL: transfer control (macro instruction)

Definitions of Terms Used in This Book

Access Method Services: A multifunction service program that defines VSAM data sets and allocates space for them, converts indexed sequential data sets to key-sequenced data sets with indexes, modifies data-set attributes in the catalog, reorganizes data sets, facilitates data portability between operating systems, creates backup copies of data sets and indexes, helps make inaccessible data sets accessible, and lists data-set records and catalog entries.

alias: A 1 to 44-character sequence that allows a user's program to refer to a nonVSAM data set or user's catalog without using the data set's or catalog's dsname. An alternate dsname for a nonVSAM data set or user's catalog.

alternate DSNAME: (*See* alias.)

alternate index: A collection of index entries organized by the alternate keys of its associated base data records.

alternate index cluster: The data and index components of an alternate index.

base catalog record: The first catalog record (control interval) that describes the VSAM object. This record contains the object's data set name, cluster name, or volume serial number in the ENTNAME field. This record also contains the header fields required for the object. The base catalog record can contain set-of-fields pointers that point to sets of fields in the base catalog record, or that point to sets of fields in extension records (vertical extension). The base catalog record's extension pointer can point to an extension record (the horizontal extension) that continues the information (set-of-fields pointers) contained in the base catalog record.

candidate volume: A direct-access storage volume that has been defined in a VSAM catalog as a VSAM volume; VSAM can automatically allocate space on this volume, as needed.

catalog: (*See* master catalog and user catalog.)

catalog recovery area: (*See* CRA.)

cluster: A combination of related VSAM data sets, identified by one name in a VSAM catalog and requiring a single DD statement. A key-sequenced data set and its index form a cluster; an entry-sequenced data set alone forms a cluster.

collating sequence: An ordering assigned to a set of items, such that any two sets in that assigned order can be collated. As used in this publication, the order defined by the System/370 8-bit code for alphabetic, numeric, and special characters.

compendium: A compendium gathers together and presents in concise form all the essential facts and details about a VSAM functional unit.

component: A named, cataloged collection of stored records. The lowest member in the data structure hierarchy. A data set contains at least one component, and the component can contain no subsets.

control area: A group of control intervals used as a unit for formatting a data set before adding records to it. Also, in a key-sequenced data set, the set of control intervals pointed to by a sequence-set index record; used by VSAM for distributing free space and for placing a sequence-set index record adjacent to its data.

control interval: A fixed-length area of auxiliary-storage space in which VSAM stores records and distributes free space. It is the unit of information transmitted to or from auxiliary storage by VSAM, some integer multiple of blocksize.

CRA: Catalog recovery area. An entry-sequenced data set that exists on each volume owned by a recoverable catalog, including the catalog volume itself. The CRA contains self-describing records as well as duplicates of catalog records that describe the volume.

data integrity: Preservation of data or programs for their intended purpose. As used in this publication, the safety of data from inadvertent destruction or alteration.

data record: A collection of items of information from the standpoint of its use in an application and not from the standpoint of the manner in which it is stored. (*See also* stored record.)

data security: Prevention of access to or use of data or programs without authorization. As used in this publication, the safety of data from unauthorized use, theft, or purposeful destruction.

data set: The major unit of data storage and retrieval in the operating system, consisting of data in a prescribed arrangement and described by control information to which the system has access. As used in this publication, a collection of fixed- or variable-length records in auxiliary storage, arranged by VSAM in key sequence or in entry sequence. (*See also* key-sequenced data set.)

data space: A storage area defined in the volume table of contents of a direct-access volume for the exclusive use of VSAM to store data sets, indexes, and catalogs.

direct access: The retrieval or storage of data by a reference to its location in a data set rather than relative to the previously retrieved or stored data.

extent: A continuous space allocated on a direct-access storage volume, reserved for a particular data space or data set.

extension record: The continuation of a catalog record that contains set-of-fields pointers and their sets of fields. Set-of-fields pointers in an extension record always point to sets of fields within the extension record. The extension record's extension pointer can point to a control interval that contains part of a set of fields too large to fit in the extension record (horizontal extension).

external procedure: A procedure that can be called by any other VSAM procedure; a procedure whose name is in the module's (assembler listing) "external symbol dictionary".

field: In a record or a control block, a specified area used for a particular category of data or control information.

free space: (*See* distributed free space.)

generation data group: A collection of data sets that are kept in chronological order; each data set is called a generation data set.

generic key: A high-order portion of a key, containing characters that identify those records that are significant for a certain application. For example, it might be desirable to retrieve all records whose keys begin with the generic key AB, regardless of the full key values.

group code: (*See* type code.)

group occurrence: (*See* set of fields.)

group occurrence pointer: (*See* set-of-fields pointer.)

horizontal extension: An extension record pointed to by a catalog record's extension field. (*See also* vertical extension.)

horizontal pointer: A pointer in an index record that gives the location of another index record in the same level that contains the next key in collating sequence; used for keyed sequential access.

index: As used in this publication, an ordered collection of pairs, each consisting of a key and a pointer, used by VSAM to sequence and locate the records of a key-sequenced data set; organized in levels of index records. (*See also* index level, index set, and sequence set.)

index entry: A key and a pointer paired together, where the key is the highest key (in compressed form) entered in an index record or contained in a data record in a control interval, and the pointer gives the location of that index record or control interval.

index set: The set of index levels above the sequence set. The index set and the sequence set together comprise the index.

index upgrade: The process of reflecting changes made to a base cluster in its associated alternate indexes.

internal procedure: A procedure that can be called only by other procedures within the module. (*See also* external procedure.)

key: One or more characters within an item of data that are used to identify it or control its use. As used in this publication, one or more consecutive characters taken from a data record, used to identify the record and establish its order with respect to other records. (*See also* key field and generic key.)

key field: A field located in the same position in each record of a data set, whose contents are used for the key of a record.

key sequence: The collating sequence of data records, determined by the value of the key field in each of the data records. May be the same as, or different from, the entry sequence of the records.

key-sequenced data set: A data set whose records are loaded in key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in the data set in key sequence by means of distributed free space. Relative byte addresses of records can change.

mass storage volume: The unit of storage in the Mass Storage System.

master catalog: A key-sequenced data set with an index containing extensive data-set and volume information that VSAM requires to locate data sets, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set, and to accumulate usage statistics for data sets.

pagespace: An amount of direct-access device space that can be used to contain temporary data sets. A pagespace is conceptually (and identified to the OS/VS2 system as) an entry-sequenced VSAM data set.

password: A unique string of characters stored in a catalog that a program, a computer operator, or a terminal user must supply to meet security requirements before a program gains access to a data set.

path: A named, logical entity composed of one or more clusters (an alternate index and its base cluster, for example).

pointer: An address or other indication of location. For example, an RBA is a pointer that gives the relative location of a data record or a control interval in the data set to which it belongs. (*See also* horizontal pointer and vertical pointer.)

portability: The ability to use VSAM data sets with different operating systems. Volumes whose data sets are cataloged in a user catalog can be demounted from storage devices of one system, moved to another system, and mounted on storage devices of that system. Individual data sets can be transported between operating systems using Access Method Services.

prime index: The index component of a key-sequenced data set that has one or more alternate indexes.

prime key: The key of reference for a base cluster, key-sequenced data set when it was loaded.

procedure: A functional unit of VSAM code that is entered only at one entry point and exits at the end of the procedure (the last line of the procedure's code). The procedure can call (transfer control, with a return to the procedure expected) other procedures within the module (internal calls) and can call other procedures in other VSAM modules (external calls). (*See also* internal procedure and external procedure.)

qualified DSNAME: A data set name in the form A.B.C.

RBA: Relative byte address. The displacement of a data record or a control interval from the beginning of the data set to which it belongs; independent of the manner in which the data set is stored.

record: (*See* index record, data record, stored record.)

relative byte address: (*See* RBA.)

relative DSNAME: A data set name in the form "dsname(n)" that identifies a generation data set.

relative record data set: A data set whose records are loaded into fixed-length slots.

relative record number: A number that identifies not only the slot, or data space, in a relative record data set but also the record occupying the slot.

reusable data set: A VSAM data set that can be reused as a work file regardless of its old contents.

security: (*See* data security.)

segment: The portion of a stored record contained within a control interval. A stored record may consist of one or more segments (see *spanned record*).

shared resources: A set of functions that permit the sharing of a pool of I/O-related control blocks, channel programs, and buffers among several VSAM data sets open at the same time.

spanned record: A logical record whose length exceeds control interval length, and as a result, crosses, or spans, one or more control intervals within a single control area.

sequence set: The lowest level of the index of a key-sequenced data set; it gives the locations of the control intervals in the data set and orders them by the key sequence of the data records they contain. The sequence set and the index set together comprise the index.

set of fields: A group of catalog record fields that contain related information. Sets of fields are referred to in the code as "group occurrences" or "GOs".

set-of-fields pointer: A field used to identify and locate a set of fields by its displacement from the beginning of the record's sets of fields (the set of fields is in the same control interval as the set-of-fields pointer) or by a control interval number (the set of fields pointer is in the base catalog record or its extension and the set of fields is in an extension record). Set-of-fields pointers are grouped by type code and are in ascending sequence by sequence number. Set-of-fields pointers are referred to in the code as "group occurrence pointers" or "GOPs".

stored record: A data record, together with its control information, as stored in auxiliary storage.

type code: A code that identifies the set-of-fields type. Type codes are referred to in the code as "group codes" or "GCs". (*See* "Field Name Dictionary" for a list of type codes.)

upgrade set: All the alternate indexes that VSAM has been instructed to update whenever there is a change to the data component of the base cluster.

user catalog: A catalog used in the same way as the master catalog, but optional and pointed to by the master catalog, and also used to lessen the contention for the master catalog and to facilitate volume portability.

vertical extension: An extension record pointed to by a set-of-fields pointer in the object's base catalog record or its horizontal extension. (*See also* base catalog record and horizontal extension.)

INDEX

For additional information about any subject listed in the index, refer to the publications listed under the same subject in *OS/VS2 Master Index of Logic,* GC28-0694.

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

SY26-3826-3

OS/VS2 Catalog Management Logic (File No. S370-30) Printed in U.S.A. SY26-3826-3

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

OS/VS2 Catalog Management Logic
SY26-3826-3

Reader's Comment Form

Your comments about this publication will help us to improve it for you. Comment in the space below, giving specific page and paragraph references whenever possible. All comments become the property of IBM.

Please do not use this form to ask technical questions about IBM systems and programs or to request copies of publications. Rather, direct such questions or requests to your local IBM representative.

If you would like a reply, please provide your name, job title, and business address (including ZIP code).

Fold on two lines, staple, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

SY26-3826-3

Fold and Staple

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:

IBM Corporation
General Products Division
Programming Publishing—Department J57
1501 California Avenue
Palo Alto, California 94304

Fold and Staple

OS/VS2 Catalog Management Logic (File No. S370-30) Printed in U.S.A. SY26-3826-3

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601