# CA MICS® Resource Management

## Space Collector Option Guide

**Release 12.9**

# Contact CA Technologies

## Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

## Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

# Contents

# Chapter 1: OVERVIEW

The CA MICS Space Collector Option (VCC) processes IBM VTOC and catalog information. It collects data pertaining to DASD storage usage and capacity and feeds that data to the CA MICS Space Analyzer Option (VCA), which stores it in the CA MICS database. The CA MICS Space Analyzer Option provides you with information and reports that help you to better manage your current storage environment by identifying problem areas in storage utilization, performing capacity planning and costing, and tracking and charging for storage usage.

The CA MICS Space Collector Option performs two batch operations that can be used to feed data into the DAILY process of the CA MICS Space Analyzer Option for analysis. Along with the standard batch job, a batch job to collect Hierarchical File System (HFS) and zSeries File System (zFS) information can also be executed.

The flow of data is displayed graphically in Figure 1-1.

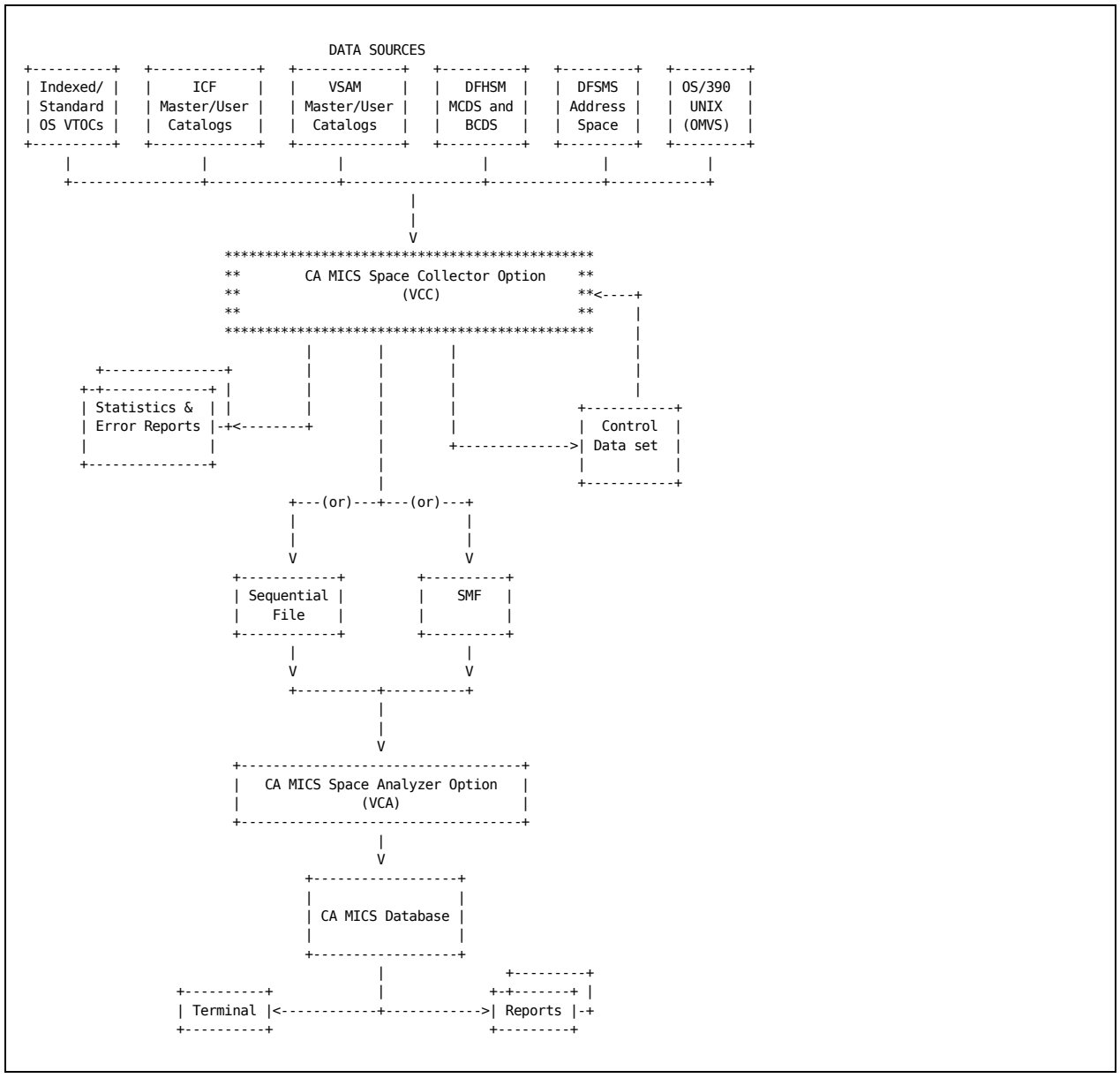



Figure 1-1. CA MICS Space Collector Option Data Flow

VCC's capabilities fall into the following categories:

- Product flexibility
- Data source facilities
- Support for IBM's Data Facility System Managed Storage (DFSMS)
- Support for IBM's Data Facility Hierarchical Storage

Manager (DFHSM)
- Report facilities
- Data collection options and features

Product Flexibility

- VCC operates on all CA MICS-supported IBM z/OS systems. Although it is designed to run below the 16-megabyte line on MVS/XA, it automatically changes its addressing mode to run in 31-bit mode when it references control blocks that reside above the 16-megabyte line.

- Because VCC has the option to reserve VTOCs, it automatically modifies itself to run as nonswappable. This is important, particularly in shared DASD environments where an address space swap out can result in several devices being held in reserve for long periods of time. Running as nonswappable, VCC ensures that device reserves are short, which keeps devices available; it also ensures that other possible CPU problems related to swap out do not occur.

- VCC keeps track of the last time it scanned a volume so that you can calculate track-days, etc. This allows you the flexibility of running VCC once a day on all volumes or once an hour on high-activity volumes.

Data Source Facilities

VCC obtains its information from the following data sources:

- Standard OS VTOCs
- Indexed VTOCs
- Standard VSAM master and user catalogs
- Master and user catalogs (VSAM and ICF)
- VSAM Volume Data Sets (VVDSs)
- The Migration Control Data Set (MCDS) and Backup Control Data Set (BCDS) maintained by DFHSM
- z/OS UNIX System Services (USS) Hierarchical File Systems

VCC reads Direct Access Storage Device volumes (DASD) to obtain information from the following IBM or vendor-equivalent device types:

- 2105
- 3340
- 3350
- 3375
- 3380 (all models)
- 3390 (all models)
- 9345 (all models)

The DASD volumes scanned must be online when VCC starts running. It does not collect information from mass storage devices and does not scan OS CVOLs because most users have converted their OS CVOL catalogs to either VSAM or ICF catalogs.

Using the data sources listed above, VCC gathers the following data:

- Storage class, management class, data class, and storage group names if available and applicable. Storage group and Management class definition information is available if DFSMS is active on the host running VCC.
- The catalog status for VTOC data sets. This optional check of whether data sets are properly cataloged assists in storage management and migration to SMS.
- All Format 1 and 3 DSCB information, including data set attributes and detailed extent information, found in indexed and nonindexed VTOCs for VSAM and VTOC data sets.
- All Format 4 DSCB information that describes the VTOC.
- Format 5 DSCB volume free space extent information. If the VTOC is indexed, the interface to IBM's Common VTOC Access Facility (CVAF) retrieves this information from the index.
- Catalog information about VSAM data sets, including how the data set was defined, data set extents on each volume for both data and index components, and alternate index information.
- Information taken from the self-describing records at the beginning of each VSAM catalog, which is used to manage the VSAM catalogs.

- Data that VCC generates when it encounters unusual situations (for example, VTOC errors, broken DSCB chains, and VSAM catalog errors) during the data collection process

- Data set name, SMS class names, and basic space allocated information for data sets migrated and backed up by DFHSM.

- Capacity planning information from DFHSM for DASD and tape volumes under control of DFHSM.

- Statistics and attributes from USS Hierarchical File Systems residing on z/OS. This includes information on the individual files within the file systems.

Support for IBM's Data Facility Hierarchical Storage Manager (DFHSM)

The VCC interface to DFHSM allows you to create an inventory of DFHSM-related information in a sequential data set or as SMF records. If you use DFHSM, this inventory collects information on inactive data that no longer resides on primary DASD as well as on backup versions that reside on DFHSM-managed tapes. This inventory allows you to measure the growth of resources allocated to DFHSM and to account to end users for this service. The CA MICS Space Analyzer Option adds inventory data to the database.

Support for z/OS UNIX Systems Services (USS)

VCC provides the ability to obtain information for each individual file within a mounted file system which resides on z/OS DASD. Once this information is processed by the CA MICS Space Analyzer Option and populated into the database, it can be used to measure the growth of POSIX compliant files and track its utilization over time.

Report Facilities

VCC provides reports that summarize the data that it reads. These reports are designed to help you track VCC operation and verify that it has read the appropriate VTOCs and catalogs. More report information is available later in this chapter.

Data Collection Options and Features

VCC offers a complete set of options and features to control and enhance its high-speed data collection process. You can specify various options by using parameters that VCC processes prior to its data collection. VCC options and features include the following:

- The ability to exploit host processors with multiple CPUs by employing subtasking methodology. This feature allows VCC to perform data collection at high speed.

- An INCLUDE or EXCLUDE list of volumes (selected by volume serial number and/or device address), data sets, catalogs, and mounted File Systems to be processed or excluded from processing.

- The ability to:

  - Specify the master catalog name, type (ICF or VSAM), and location.

  - Write data records to SMF or to a user-defined data set.

  - Specify a user-defined SMF record number for the output records.

  - Specify the number of subtasks used in performing data collection.

  - Selectively generate reports.

  - Specify weather to collect information from:
    - VTOCs
    - ICF and VSAM Catalogs
    - Mounted File Systems
    - Hierarchical Storage Manager (DFSMS)

  - Specify internal VCC performance tuning parameters.

  - Detect and report both catalog structure errors and VTOCs that are too small.

Most parameters have defaults. If VCC runs without a parameter list, it will record all VTOC data on all online, resident, and ready DASD devices. It will also scan all supported catalogs on those volumes that are online and ready.

This section contains the following topics:



# 1.1 Primary Areas of Application

The CA MICS Space Collector Option provides data and reporting capabilities that aid you in managing complex DASD systems in the following areas:

- Performance--You can monitor DASD space usage, including data set allocation, free space fragmentation, and alternate track assignments.
- Capacity Planning--You can extract MVS VTOC and catalog data that the CA MICS Space Analyzer Option can use to monitor DASD demand. This facilitates growth-trend projections at both the division and department levels.
- Accounting and Chargeback--VCC provides DASD usage data through VCA to CA MICS Accounting and Chargeback to facilitate accounting for all or selected system users.
- SMS Migration--VCC can optionally identify uncataloged data sets. This facilitates migration to DFSMS because it allows you to identify data sets that must be either deleted or cataloged prior to volume conversion for SMS.

VCC also contributes to the overall benefits available from the CA MICS system by:

- Improving day-to-day IS management by allowing managers to track DASD usage and plan corrective actions to ensure smooth operations
- Reducing risk to the enterprise by providing audit and control capabilities for all DASD use
- Controlling or reducing costs by:
  - allowing you to equitably charge users for the amount of DASD they use
  - identifying, in conjunction with other operating system management products, unused resources to substantially reduce the products' associated costs
  - eliminating the costs associated with maintaining other, separate DASD reporting products
- Improving return on the IS investment by:
  - providing a basis for operating system and hardware resource usage chargeback either through CA MICS Accounting and Chargeback or a basic user-developed costing process
  - improving your ability to understand, report on, control, and, if necessary, reduce DASD expenses
- Increasing productivity by:
  - allowing less experienced users to interpret report results and use database information
  - reducing the task of coordinating information from multiple sources so that you can report or analyze information at the enterprise rather than the system level
  - reducing the large quantities of DASD usage data to a manageable and easily used format

o Improving planning by:

- allowing you to perform trend analysis through the use of summarized historical data

- collecting DFSMS data that can be used to plan or monitor DFSMS implementation

- collecting details on the POSIX files and

directories within a Hierarchical File System.

## 1.2 Reporting and Inquiry Facilities

The CA MICS Space Collector Option generates reports during its data collection process. These reports provide statistics and log unusual situations that were encountered.

VCC produces the following information:

| Report | Function |
|---|---|
| VTOC Scan Log | Logs the events that occurred during the standard DASD data collection process. If abnormal situations were encountered, the log may contain additional messages to assist with problem determination and resolution. |
| Message Log | Logs the events that occurred during the file system data collection process. If abnormal situations were encountered, the log may contain additional messages to assist with problem determination and resolution. |
| Catalog Scan | Logs the events that occurred for the standard VCC data collection process while obtaining data for ICF or VSAM catalogs. If abnormal situations were encountered, the log may contain additional messages to assist with problem determination and resolution. |

| | |
|---|---|
| VCC Run Status | Contains a list of statistics obtained by the standard VCC data collection, as well as summary reports on DASD volumes, DFSMS, Storage groups, and System Catalogs. The statistics include:<br>o The number of DASD devices in the I/O configuration.<br>o The number of catalogs processed.<br>o The elapsed times of the VTOC and catalog scans.<br>o The count of records written.<br>o The previous data collection date and time.<br>o The elapsed time of the current data collection.<br>o Other information useful in managing volumes, System Catalogs, and DFSMS Storage Groups. |
| VCC Run Statistics | Contains a summary of the VCC run statistics for the Hierarchical File Systems data collection. The statistics include:<br>o The count of records written.<br>o The number of HFS data sets monitored.<br>o The number of POSIX files processed.<br>o The previous data collection date and time.<br>o Collection elapsed and CPU time.<br>o Other useful statistics. |
| DFHSM Interface | Contains statistics from the DFHSM interface of the standard VCC collection. |

# 1.3 Requirements

The CA MICS Space Collector Option runs in all of the following operating system environments using these distributed load modules:

o OS/390 (OS/390 through V2R10)
o z/OS (z/OS through V1R13)

The CA MICS Space Collector Option must run from an APF-authorized library to open and read ICF catalogs, enqueue VTOCs, optionally write records to SMF, and run as nonswappable. Because it runs multiple subtasks for efficiency, it may require a larger region size than your standard default. The number of subtasks can be user-controlled to allow flexibility in environments that are constrained in virtual storage.

The CA MICS Space Collector Option typically requires less than one MB of memory below the 16 MB line. Therefore, we recommend running it with the MINIMUM value of REGION=1M or REGION=1024K.

With large DASD configurations, you will want to run with many subtasks (for example, VTOCSUBT=20, VSAMSUBT=10).

# 1.4 Files Overview

The CA MICS Space Collector Option records data in a standard SMF-style format that can be written to either the standard SMF data set or to a user-defined data set. The records have a segmented structure and each segment is identified by an offset, length, and number of segment occurrences.

The segments created are the following:

- DFSMS configuration segment
- VCC Control information segment
- Non-VSAM data set segment
- The VSAM data set segment
- The VTOC/volume segment
- The Hierarchical File System (HFS) segment
- The HFS Directory entry segment
- The User defined segment

Each segment may or may not exist on a one-to-one basis for each record generated. See Figure 1-1 for a graphical representation of the data flow from VCC.

# 1.5 Product Prerequisites

The CA MICS Space Collector Option runs on any operating system from MVS/ESA (OS/390 V2R10) through z/OS V1R13. Both the standard and the HFS VCC collection programs must run from an APF-authorized library in order to perform certain authorized system functions, such as opening system catalogs, writing records to SMF, and running as non-swappable.

Since the CA MICS Space Collector Option runs multiple subtasks for efficiency, it may require a larger region size than your standard default. The number of subtasks can be user-controlled to allow flexibility in environments that are virtual storage constrained.

The VCC HFS collection batch job must be authorized to read all of the files created within a Hierarchical File System. To do this, VCC uses UID 0. Ensure that your External Security Product allows VCC to switch to UID 0, and has read access to the Hierarchical File Systems. In order to function properly, VCC HFS collection must have authority to access the BPX.SUPERUSER resource in the FACILITY class. This will allow the required SETUID(0) to function successfully.

# Chapter 2: PLANNING

This chapter deals with planning issues that must be considered before VCC is installed at your site. We recommend that you read and thoroughly understand this chapter and Chapter 3, Usage Guidelines, before you attempt to install and run VCC.

This section contains the following topics:



## 2.1 Software Dependencies

VCC is designed to have few dependencies on the operating system. However, in order for VCC to operate correctly, some dependencies on the operating system do exist and you should be aware of them in case there is a conflict.

The VCC operating system dependencies are the following:

- OS/390 UCB control block. VCC uses the UCBSCAN service to obtain information on all the online DASD volumes. All UCBs are pinned so that any dynamically added devices are not deleted during VCC execution. Once processing of the UCB is completed the UCB is unpinned.

- The CAXWA control block chain for all open VSAM/ICF catalog. If a master catalog name (VSAM or ICF) and its volume serial are not supplied, this information is obtained by scanning the CAXWA control block chain.

- SVC 26 (LOCATE) is used to locate VSAM information. This SVC is used to allocate and open catalogs (VSAM or ICF) before they are opened as data sets. VCC depends upon the format of the SVC 26 parameter list.

- The STEPCAT flag is turned on in the PCCB that is created when a catalog is opened as a catalog through the use of SVC 26. This is done so that serialization problems do not arise when multiple subtasks attempt to open and read catalogs as data sets. After a catalog has been processed by VCC, this bit is reset (turned off). This eliminates the requirement to have STEPCATs if VSAM catalogs are being processed as data sets.

- VCC optionally records to SMF using the SMFWTM macro.

- To collect data regarding data sets managed by DFHSM, the ARCUTIL interface program is required. This program was delivered by IBM in Version 2.5 of DFHSM and certain enabling PTFs support this function at Version 2.4 of DFHSM. Check with IBM Customer Support for the PTF numbers required at the 2.4 level of DFHSM. This ARCUTIL interface is the same technique used by the DCOLLECT function of IDCAMS available at DFP Version 3.2 and above.

- VCC calls various OS/390 UNIX System Services (USS) Interface modules provided by IBM to obtain information on Hierarchial File Systems residing on OS/390.

## 2.2 Supported MVS Data Structures

VCC supports several MVS structures as input data sources. They are explained in the following subsections:

1 - Standard VTOC Support
2 - Indexed VTOC Support
3 - VSAM Catalog Support
4 - Integrated Catalog Facility (ICF) Support
5 - The SMS Address Space
6 - DFHSM Migration Control Data Set (MCDS)
7 - DFHSM Backup Control Data Set (BCDS)
8 - z/OS UNIX System Services (USS) Support

NOTE: OS CVOLs are not supported.

### 2.2.1 Standard VTOC Support

VCC supports standard VTOCs. A standard VTOC is defined as being a volume table of contents that does not have an index associated with it.

VCC scans a VTOC using VSAM chained scheduling, reading DSCBs a track at a time, into an I/O buffer. All information in the following DSCBs are placed into the VCC record:

- Format 1 DSCB - identifies a data set on the device
- Format 2 DSCB - ISAM index information
- Format 3 DSCB - data set extent information
- Format 4 DSCB - VTOC information
- Format 5 DSCB - volume free space extent information

The first DSCB in a standard VTOC is the Format 4 DSCB. It describes the extents of the VTOC, characteristics of the device on which it resides, and information about the DSCBs in the VTOC (number of free DSCBs and a CCHHR pointer to the last Format 1 DSCB). It also contains VSAM ownership timestamps (not applicable to ICF catalogs). There is only one Format 4 DSCB in a standard OS VTOC.

VCC uses the number of available DSCBs to calculate the termination point of its VTOC SCAN (total DSCBs minus available DSCBs = total used DSCBs).

The next DSCB in a standard VTOC is the Format 5 DSCB. Each Format 5 DSCB describes up to 19 free space extents on the volume. There is also a CCHHR pointer to the next Format 5 DSCB.

Following the above two DSCBs, the other DSCBs can occur in any order. The Format 1 DSCB can either be self-contained (if it has less than four extents), can point to a Format 3 DSCB (if the data set has more than three extents), or can point to a Format 2 DSCB (ISAM only) which then can point to a Format 3 DSCB.

### 2.2.2 Indexed VTOC Support

VCC also scans VTOCs that have an index. This is valid only if the VTOC has been converted to have an index associated with it.

An indexed VTOC is comprised of two elements: the VTOC itself and its associated index. The VTOC is structured in the same way that a standard VTOC is structured with the following exceptions:

- The Format 4 DSCB will no longer contain the CCHHR pointer to the last-used Format 1 DSCB and the number of free DSCBs. This information is kept in the index and is retrieved using the CVAF interface.

- Volume free space extents are no longer kept in the Format 5 DSCB. There is a record in the index that contains a bit map of the used tracks on the volume. There is still one Format 5 DSCB in the VTOC. It follows the Format 4 DSCB and contains no information. Its sole purpose is to provide a means by which an indexed VTOC can be converted back to a standard VTOC.

### 2.2.3 VSAM Catalog Support

VCC supports both master and user catalogs. It scans the low key-range portion of a VSAM catalog to collect and records the following:

- VSAM catalog self-describing information
- VSAM data set information
- VSAM alternate index information

VSAM Catalog Overview

A VSAM master or user catalog is a VSAM key-sequenced data set (KSDS) that is divided into two sections: the low key-range and the high key-range. Each control interval (physical record) in a VSAM catalog is 512 bytes long. Each of the logical records in the low key-range portion of a catalog are one control interval in length. The low key-range portion of a catalog contains detailed information about the following:

- VSAM data sets
- VSAM alternate indexes
- VSAM paths
- VSAM data spaces
- DASD volumes owned by VSAM
- VTOC data set volume pointers
- OS/CVOL pointers
- Alias associations with OS/CVOLs or VSAM user catalogs

The logical records in the high key-range portion of a catalog are 47 bytes long and there can be several of these records in one control interval. These records are referred to as "truename" catalog records. They are comprised of the following:

- A 44-byte data set name of a data set, cluster, user catalog, VTOC data set, alternate index, path, alias (blank-filled on the right), or volume serial number (zero-filled on the right). This is the key.

- A three-byte control interval number of the catalog record in the low key-range portion of the catalog that describes the object identified above.

A catalog is therefore accessed by doing a keyed read specifying the data set name of the object being located. The truename record is returned. VSAM then uses the control interval pointer in the truename record to retrieve the remainder of the information on the object being located.

For more information on VSAM catalog structure, see the IBM Publication SY26-3826, OS/VS2 MVS Catalog Management Logic.

## 2.2.4 Integrated Catalog Facility (ICF) Support

VCC supports the Integrated Catalog Facility (ICF). It reads ICF Basic Catalog Structures (BCSs) and VSAM Volume Data Sets (VVDSs) to collect and record the following information:

- VSAM data set information
- VSAM alternate index information

ICF Catalog Structure Overview

The ICF catalog structure is comprised of two components: the BCS and the VVDS.

The BCS

A Basic Catalog Structure is a standard VSAM key-sequenced data set with spanned records. It contains the volume, data set security, ownership, and association information for VSAM data sets and ICF catalogs. For VTOC data sets, the BCS contains extent information.

The VVDS

A VSAM Volume Data Set is a standard VSAM entry-sequenced data set with 4K control intervals. It physically resides on the same volume as the VSAM data sets that are defined in it, and contains the dynamic characteristics of these data sets. Up to 36 BCSs can point to one VVDS; therefore the concept of VSAM volume ownership no longer exists in the ICF catalog structure. A VVDS is identified by the reserved data set name SYS1.VVDS.Vvolser. If you use the DFSMS (System Managed Storage) facility, the VVDS will also contain SMS related information for VTOC data sets residing on the volume.

Differences Between ICF Catalogs and VSAM Catalogs

In addition to the structural differences between ICF catalogs and VSAM catalogs, the following additional differences should be noted:

- ICF catalogs can be defined with all attributes that relate to VSAM key-sequenced data sets. This includes tuning parameters that were not available for VSAM catalogs.
- VSAM data spaces are no longer supported in ICF catalogs.
- As noted previously, the concept of volume ownership no longer exists in the ICF catalog structure.
- Truename records no longer exist, resulting in faster catalog searches.
- Multiple ICF catalogs can reside on the same volume.
- ICF catalogs can coexist with VSAM catalogs and OS CVOLs. The system master catalog can be either a VSAM catalog or an ICF catalog and can point to either VSAM or ICF user catalogs.

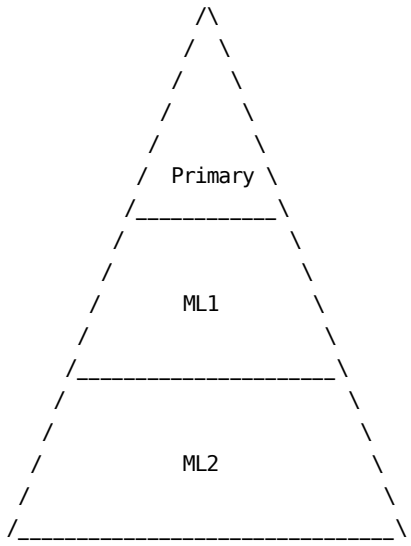
## 2.2.5 The SMS Address Space

If DFSMS (or SMS) is active at execution time, VCC will attempt to obtain the storage group name and status for a given volume by means of a subsystem interface call to SMS. The storage groups found are reported on the VCC Run Status Report (refer to Figure 3-3) and placed in the volume summary record written along with the volume serial number. As storage administrators move toward management and reporting at the volume pool level and away from individual volume reporting, storage group name becomes the preferred organizing principle.

## 2.2.6 DFHSM Migration Control Data Set (MCDS)

VCC can optionally collect basic accounting and capacity planning information about data sets that have been migrated from standard VTOCs and moved to compressed locations in the hierarchy of storage managed by DFHSM. Locator information about migrated data sets is stored in the DFHSM Migration Control Data Set (MCDS). VCC serves as a "driver" for the collection of data from the MCDS. The actual data collection from DFHSM data structures is performed by a DFHSM-supplied utility program called ARCUTIL that was delivered in DFHSM Version 2.5 to protect user-written utilities from changes in the record layouts in the MCDS and BCDS. VCC "drives" ARCUTIL by ATTACHing it as a subtask and intercepting its output to squeeze out padding blanks and to add important time-stamp and control information.

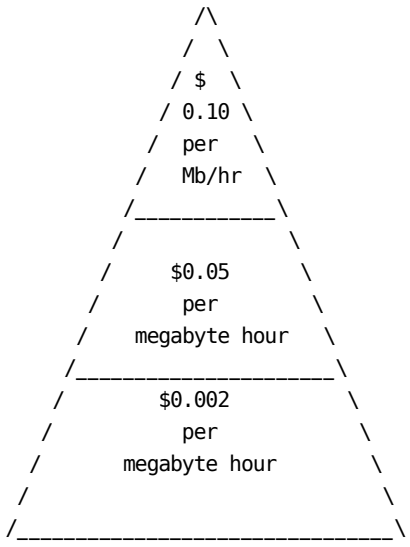
Data in a DFHSM environment can be conceptualized as pyramidal in shape. This pyramid is meant to convey a hierarchy of storage ranging from PRIMARY or migration level 0 to migration levels 1 and 2. For simplicity, we abbreviate "migration level" to "ML" and thus, it is common practice to describe data as residing on primary, ML1, or ML2 storage. Primary is the normal location of data directly accessible by applications. When a data set has not been used for some specified period, HSM migrates it to ML1 storage volumes, where the data is usually stored in a compacted form and is not directly accessible. It can be recalled quickly to primary storage when needed. Finally, when installation thresholds have been reached, HSM will migrate data from ML1 DASD to ML2 storage. ML2 storage is usually a densely-packed tape cartridge.




In a typical DFHSM environment, the amount of data migrated through the above hierarchy could be presented as the above pyramid. For example, if 400 gigabytes of primary storage is available for active data, then perhaps a much larger quantity of data is inactive and has been migrated to less expensive levels of the hierarchy.

The VCC driver for DFHSM data collection anticipates a large quantity of inactive data and is designed for optimum execution time performance and optimum data compression of the output file it creates.

One of the principal benefits of integrating DFHSM data collection into VCC and subsequently into CA MICS is that there is no longer any need for capacity planners or accounting specialists to view DFHSM-controlled data as a special case. For example, using VCC in conjunction with CA MICS products, you can establish different pricing for each level of the storage hierarchy. Using the pyramid again, you could conceive a rate structure that reflects the relative cost of each layer, as shown in the following example:



## 2.2.7 DFHSM Backup Control Data Set (BCDS)

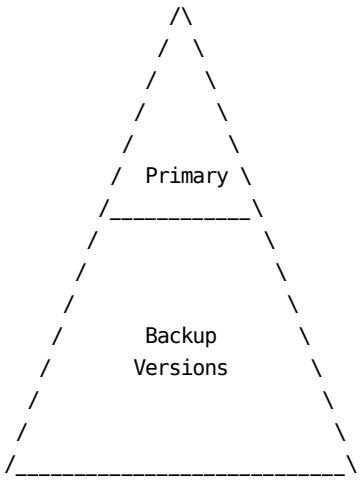
VCC can optionally collect basic accounting and capacity planning information about backup versions of data sets. When DFHSM performs automatic backup of data sets, the BCDS is updated with information about each backup copy or backup version created on backup volumes managed by DFHSM.

The pyramid forms a useful analogy for considering the potential amount of information processed by the VCC-DFHSM interface for backup versions. Consider this pattern of usage for typical TSO users and the backup rules associated with a "STANDARD" management class:

- 25 percent of a given user's data sets are changed each day.
- An HSM rule states data sets are only to be backed up if they have been changed.
- A management class definition states that 5 backup versions are to be kept.

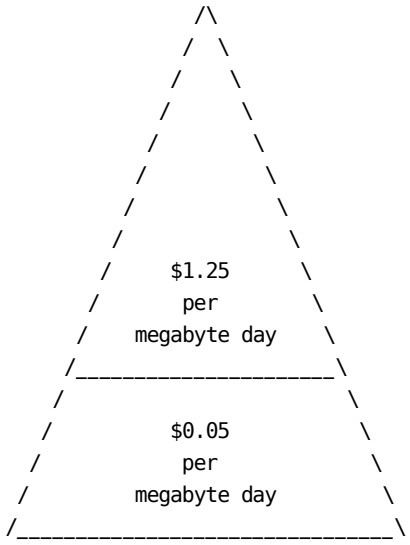
Thus, for an active data set that is changed every day, the BCDS will contain information for 5 backup versions of this data set in our hypothetical example above.

Expanding this scenario to an entire DASD complex, there might be the following kind of relationship in terms of the relative number of data sets:



The VCC driver for DFHSM data collection anticipates a large quantity of backup versions. It is designed for optimum execution time performance and optimum data compression of the output file it creates. Finally, you can collect data about migrated data sets without collecting data about backup versions (or vice versa).

Using VCC in conjunction with other CA MICS products, you can establish different pricing for storage occupied for backup versions. Because the backup versions typically exist on high-density tape cartridges, the relative charge for storage occupancy (that is, amount of space times duration or megabyte-days) is undoubtedly less for backup versions than for primary DASD storage occupancy. An example follows:




Capacity planners can exploit this data by analyzing the growth in backup versions over time, projecting the trend by division or department using normal CA MICS techniques and SAS procedures.

## 2.2.8 UNIX System Services (USS) Support

The CA MICS Space Collector Option (VCC) can optionally collect information for UNIX System Services (USS) file systems. This information can be fed into the CA MICS Space Analyzer Option (VCA). The support allows for the collection and reporting of statistics, as well as the tracking of actual space utilization of the USS files compared to the file systems' allocated DASD space.

VCC provides a separate batch job, located in member sharedprefix.MICS.CNTL(VCCRUNH), to collect file system information. The data collected from this job can be fed into VCA for further processing and analysis. It collects file systems information for all the file systems mounted to the z/OS system that the job runs on. If the AUTOMOUNT facility is in use, the directory designated for automount processing will be scanned. This results in the associated file system and any file systems under that directory structure to be automatically mounted, if they are not already mounted. The file systems that are unknown to z/OS or are unmounted will not be scanned. In the event that your site has multiple z/OS images using various file systems which may not be mounted on all z/OS images, it is recommended that you run VCC on all z/OS images that have file systems mounted. For a discussion on shared DASD and z/OS file systems considerations, see section 2.5.3.

Because of the way USS accesses file systems, plan to monitor how long VCC takes to collect information on these file systems initially. Usually the root file system takes the longest time to process. Therefore, you may want to weigh the benefit of multiple scans during the day with the performance cost. Optionally, you may exclude some file systems from collection.

# 2.3 SMF Requirements

VCC writes variable length records to either a physical sequential file or to SMF. If DFHSM is implemented and the VCC-DFHSM data collection interface is selected, the VCC-DFHSM interface will likewise write variable length records to either a (different) sequential file or as a (different) SMF record.

Finally, for both standard VCC records and VCC-DFHSM records written to their respective output data sets, a hybrid approach can be effected by copying (DISP=MOD) or concatenating these data sets into the dumped SMF data set.

If direct recording to SMF is chosen:

- User record numbers must be between 128 and 255.
- SYS1.PARMLIB(SMFPRMxx) must allow the user record to be written.
- SMF recording of VCC data is independent of SMF recording of VCC-DFHSM data.
- If both types of data are being logged to SMF, the record numbers specified must be different.

More specific information on SMF requirements is available in the following subsections:

1 - SMF Requirements (VCC)
2 - SMF Requirements (VCC-DFHSM)

### 2.3.1 SMF Requirements (VCC)

VCC supports SMF as a recording medium. It produces a segmented record that follows the SMF record segmentation conventions (for example, as used by SMF type 30 and RMF type 70-series records) where a segment specifies the offset, length, and number of occurrences of the next segments.

The choice of using SMF as a recording facility for VCC depends solely on the size of your installation and the amount of DASD data VCC is expected to collect. VCC generates one SMF record for every:

- Format 1 DSCB encountered in a VTOC (every data set).
- DASD volume scanned.
- VSAM data set found in a catalog.
- VSAM alternate index found in a catalog.
- VSAM data set found in a VVDS (ICF catalogs only).
- VVDS information for VTOC data sets managed by DFSMS.
- VSAM alternate index found in a VVDS (ICF catalogs only).
- VSAM data set found in a VVDS (ICF catalogs only) and VVDS information for VTOC data sets managed by DFSMS are repeated if a VSAM Volume Record (VVR) representing a data set fills up due to extents and needs to have a secondary VVR built.
- If a VSAM data set's index component has the IMBED option specified, a separate record is generated by VCC. This is because the index's sequence set resides in the first track of each data component control area. That space is described by a separate record in the VVDS, causing another record to be generated by VCC. This is duplicate recording of space utilization because the sequence set space is also described in the data component extents, but is unique in other respects (for example, low RBA, high RBA, etc.). The duplicate recording of space information is ignored by the CA MICS Space Analyzer Option.
- VSAM master and user catalog.

Let's assume that you have the following:

```
10000 VTOC data sets
  100 VSAM clusters and alternate indexes
   10 ICF BCSs including the master catalog
   60 VVDSs (one on each volume with a VSAM data set)
   85 DASD volumes
```

VCC will then generate at least:

```
10000+(100*3)+10+85=10395
```

SMF records. Note that VSAM is counted three times for ICF because there is a VSAM data set entry in the BCS, as well as a VVDS entry and a VTOC entry.

The average size of each SMF record is variable because of the variance in the number of extents for each data set and the variance in volume free space fragmentation from one site to the next. In testing the product, the average record size was found to be approximately 235 bytes. Using this number, VCC records generated by the above site will consume about 2.4 million bytes of DASD space. Using 4K physical SMF records, this is equivalent to approximately 153 tracks on a 3350, or about five 3350 cylinders.

A consideration that may help you decide between recording VCC data to SMF or to a data set is the speed at which this data will be sent to SMF. VCC is capable of generating 10,000 SMF records to a data set in a little under 20 seconds of elapsed time running on a lightly loaded 3033U processor. If this same amount of data is sent to SMF, a period of system-wide degradation can occur due to SMF buffer shortages, especially during prime shift peak hours.

## 2.3.2 SMF Requirements (VCC-DFHSM)

The VCC interface for DFHSM data collection generates variable length records that can be written to the VCCHSM data set or logged to SMF. The records vary in size from 63 to 126 bytes. Variable length fields such as SMS class names and dsname are stored as variable length strings with no padded blanks.

Due to the high volume of data potentially found for data sets under DFHSM management, we recommend that you do NOT log data from this interface directly to SMF. You should use the VCCHSM data set to receive the records from this interface. One record will be written for each:

- Migrated data set on ML1 or ML2 storage as defined in the MCDS at the time of data collection.
- Backup version of any data set as defined in the BCDS at the time of data collection.

The above items represent the vast majority of data written by this interface. In addition, this interface writes one record (called a DASD planning record) for each:

- Primary volume (migration level 0) for each day there was HSM activity.
- Migration level 1 volume for each day there was DFHSM activity.

Last in terms of number of records (bytes) written by this interface is a series of three tape capacity planning records. The VCC-DFHSM interface writes one tape capacity planning record for each of the following types of tape:

- Migration level 2 tapes
- Incremental backup tapes
- Full volume backup tapes

In summary, consider the following example as an aid in understanding the number of records put out by this interface:

100,000 "active" permanent data sets on primary DASD volumes

20 percent of these data sets are NOT backed up using DFHSM incremental backup or HBACK commands

Of the remaining 80,000 data sets on primary DASD, DFHSM maintains an average of 5 backup versions for each data set

(B = 80,000 * 5 = 400,000)

50,000 data sets are migrated to ML1 volumes

(ML1 = 50,000)

200,000 inactive data sets have been migrated to ML2 volumes

(ML2 = 200,000)

200 DASD volumes have DFHSM activity each day and MIGRATIONCLEANUPDAYS(5) is specified in the DFHSM SETSYS command

(C= 200 * 5 = 1000)

Three tape inventory records are always written

(T = 3)

The summation of the above figures provides the number of records produced to the VCCHSM data set or to SMF in a given inventory:

```
TOTAL = B + ML1 + ML2 + C + T

TOTAL = 400,000
      + 50,000
      +200,000
      +  1,000
      +      3
      +      1   (Trailer record)
      --------
       651,004   records written to VCCHSM (or to SMF)
```

# 2.4 CA MICS Considerations

The CA MICS Space Collector Option (VCC) is intended to be run at least once per day on each SYSID in a shared DASD configuration with care taken to use EXCLUDEVOL on shared volumes to prevent duplicate data from being introduced. While there are many operational choices you can make (as discussed in Chapter 3), there are various considerations for CA MICS processing of this data collected by VCC. The data VCC creates will be processed by VCA.

The next sections discuss the CA MICS considerations for each of the above products from the point of view of the collector.

More specific information on CA MICS considerations is available in the following subsections:

1 - CA MICS Considerations (VCC)
2 - CA MICS Considerations (VCC-DFHSM)
3 - CA MICS Considerations (VCC-HFS)

## 2.4.1 CA MICS Considerations (VCC)

VCC is intended to be run once per day to collect data for CA MICS applications. The SMF-like records written to the ddname VCCDATA (or SMF) will be fed into the CA MICS DAILY job, in particular, the CA MICS DAY090 step, which will organize the VCC data into several different VCA files. These VCA files can later be reported on with the VCA reporting facility and SAS. These same VCA files will also be used as inputs to the following CA MICS Management Support Applications:

- CA MICS Accounting and Chargeback
- CA MICS Capacity Planner
- CA MICS StorageMate

Run-time parameters allow VCC to collect only volume level statistics (DATAINFO=N) or catalog status of VTOC data sets (CATCHECK=Y). These runs should be distinctly different from the normal production use of VCC. For DATAINFO=N applications where only one record per VOLSER is produced, the probable audience for any reports produced from this data is the storage administration staff, because accounting applications are impossible without data set information. Accordingly, it is important that you set up this type of VCC application with VCCDATA and VCCNTROL files that are different from the production run. CATCHECK processing, on the other hand, could use the same VCCDATA and VCCNTROL files as the production use of VCC but should be scheduled infrequently due to processing time requirements. Chapter 3 discusses these options in greater detail.

VCC must be run well before the CA MICS DAILY job in order to allow enough time for a rerun in case there are problems. If VCC data goes to SMF and VCC cannot be rerun before CA MICS daily processing, only those data sets on volumes that were COMPLETELY processed by VCC before the problems occurred will be handled by CA MICS.

If the VCC SMF record number changes (the default is 198), the CA MICS Space Analyzer Option must be made aware of this fact at the time the change is made. If this is not done, VCC data will be lost.

To inform CA MICS about a change in the VCC SMF record number (even if you log records to the VCCDATA file and not to the SYS1.MANx structures), you must change the INPUTSOURCE value in sharedprefix.MICS.GENLIB(VCAGENIN) and run VCACGEN.

## 2.4.2 CA MICS Considerations (VCC-DFHSM)

The "production" run of VCC that collects data for primary DASD from VTOCs and catalogs can be extended to collect data for inactive data sets and backup versions through use of the DFHSM Inventory part of the CA MICS Space Analyzer Option. This provides a VCC collection module to obtain information about objects under the direct control of DFHSM.

The SMF-like records emanating from the VCC-DFHSM interface are written to either ddname VCCHSM (or SMF) and will be fed into the CA MICS DAILY job, in particular, the DAY095 step that will organize the VCC-DFHSM data into several different DFHSM files. These DFHSM files can later be reported on with the user-written SAS programs, or MICF. These same DFHSM files will also be used as inputs to these CA MICS Management Support Applications:

- CA MICS Accounting and Chargeback
- CA MICS StorageMate

Run-time parameters allow VCC to collect only HSM data (HSM=Y with VTOCSUBT=0) or everything in the same run. The important consideration in a shared-DASD environment is to collect DFHSM data only once per day from only one SYSID. See the IBM publication SH35-0084-4 or above, DFHSM Installation and Customization Guide, for information about DFHSM operation in a shared-DASD environment. Chapter 3 of this guide discusses each HSM-related option in greater detail.

The DFHSM data created by VCC must be created before the CA MICS DAILY run in order to allow enough time for a rerun in case there are problems. If the DFHSM data goes to SMF and VCC cannot be rerun before CA MICS daily processing, this batch of HSM data would be treated as an incomplete inventory.

If the VCC SMF record number for DFHSM data changes (the default is 197), VCA must be made aware of this fact at the time the change is made. If this is not done, VCC-DFHSM data will be lost.

To inform CA MICS about a change in the VCC SMF record number for DFHSM (even if you log records to the VCCHSM file and not to the SYS1.MANx structures), you must change the INPUTSOURCE value in sharedprefix.MICS.GENLIB(HSMGENIN) and run HSMCGEN.

## 2.4.3 CA MICS Considerations (VCC-HFS)

The CA MICS Space Collector Option (VCC) can collect directory and file information for file systems that reside on z/OS.

If collection of USS file systems information is desired, a separate sample job VCCRUNH is provided in sharedprefix.MICS.CNTL data set.

This batch job can run simultaneously with the standard VCCRUN job. The records produced for the file systems follow the standards for creating SMF records and can be written to the ddname VCCDATA. This information can be fed into the CA MICS DAILY job by concatenating it to the VCCDATA generated by the standard VCCRUN job, if SMF recording is not specified. If this is not done, VCC HFS scan data will be lost.

The VCC SMF record number for HFS data must match the SMF record number specified for non-VSAM and VSAM VCC data. If the VCC SMF record number for HFS, VSAM and non-VSAM data changes (the default is 198), VCA must be made aware of this change. If this is not done,
VCC-HFS data will be lost.

To implement a change to the VCC SMF record number for HFS (even if you do not specify SMF recording), perform the following steps:

1. Edit sharedprefix.MICS.GENLIB(VCAGENIN)

2. Add or modify the SMF record number in the INPUTSOURCE statement

3. Submit sharedprefix.MICS.CNTL(VCACGEN).

Once processed by the CA MICS Space Analyzer Option, the information about file systems can be used in the following CA MICS Management Support Applications:

- CA MICS Accounting and Chargeback Option (Not Recommended)

- CA MICS StorageMate Option

Runtime parameters allow for the reduction of the information provided for the file systems. Specific file systems or volumes may be excluded from data collection.

Because file systems are logically unavailable to VCC if they are not mounted on any z/OS system, VCC should run on z/OS images where mounted.

# 2.5 Special Considerations

This section describes the following special considerations that must be planned for before VCC is installed:

1 - Authorization Considerations
2 - Security Package Considerations
3 - Shared DASD Considerations
4 - DASD Reserve Considerations
5 - Multi-Image Manager (MIM) Considerations

## 2.5.1 Authorization Considerations

Because VCC allocates and opens for input VSAM catalogs, ICF BCSs and VVDSs, and ENQs VTOCs, it must run as an APF-authorized program. All of the VCC modules are link-edited with an authorization code of 1 and placed in the CA MICS LOAD distribution library. To make sure that VCC runs as an authorized program, you must perform one of the following:

- Add the CA MICS LOAD distribution library name to the list of authorized libraries. You can do this by adding the CA MICS LOAD distribution library name to the IEAAPFxx member in the SYS1.PARMLIB that is used for IPL.

- Copy the VCC modules from the CA MICS LOAD distribution library to a program products library whose name is already in the APF.

- Copy the VCC modules from the CA MICS LOAD distribution library to an authorized link list library. All link list libraries are automatically authorized unless you change this default using the LNKAUTH parameter in SYS1.PARMLIB(IEASYS00). SYS1.LPALIB is authorized only during IPL, otherwise it must be specifically placed in the APF or link list to be authorized.

The copy operations above can be done using the SAS PDSCOPY facility (which is recommended), the IBM IEBCOPY facility (if the blocking attribute of the two libraries is the same), or the IBM linkage editor.

## 2.5.2 Security Package Considerations

VCC allocates and opens ICF and VSAM catalogs, VVDSs, and OS/VTOCs. Enterprise Security packages such as IBM's RACF, CA Top Secret, and CA ACF2 provide open exits that match the name of the data set being opened against a list of restricted names defined by your security officer. Quite often names of the aforementioned special data sets are in a list of secured names, which means that normal VSAM password protection functions are bypassed and handled instead by the security package.

Note: To avoid security violations, you must thoroughly research the security in force for VSAM and ICF catalogs and VTOCs (if applicable) to ensure that the user ID under which the VCC scan is run has READ access to the names of the catalogs and VTOCs that VCC will be scanning. The user ID must also have read access to all PDSEs on which it will be collecting data. If HSM=Y is coded in the runtime parameters, the user ID under which the scan is run must have read access to the MCDS and BCDS that the scan job is accessing.

VCC uses the Callable Assembler Interface to UNIX System Services to scan the Hierarchial File Systems. In order to access UNIX System Services, the userid assigned to the batch job must be defined to UNIX System Services security. The userid must be assigned a UNIX userid, a groupid, and a home directory. See the appropriate security product's manuals for additional information.

Prior to scanning the mounted Hierarchial File Systems, VCC issues the function SETUID(0) (set USERID to SUPERUSER) in order to ensure having read access to all the files within the HFS structure. In order to function properly, VCC must have authority to the BPX.SUPERUSER resource in the FACILITY class. This will allow the SETUID(0) to function successfully.

For the IBM z/OS Security Server RACF, see the IBM manual "UNIX System Services Planning" Chapter 16, (Establishing UNIX Security) for additional information. Review the sections of chapter 16 relating to BPX.SUPERUSER, Assigning Superuser Attributes, Using UNIXPRIV Class Profiles, and Setting up The BPX.* FACILITY Class Profiles BPX.SUPERUSER. For CA ACF2, the userid that is assigned to the batch job must be defined to CA ACF2 Unix System Services Security. See the CA ACF2 Administrator Guide, chapter 21 (z/OS UNIX System Services Support) for information about defining a userid and setting up the ability to execute the setuid(0)

function.

For CA Top Secret, the userid that is assigned to the batch job must be defined to CA Top Secret Unix System Services Security. See the CA Top Secret documentation for information about defining a userid and setting up the ability to execute the setuid(0) function.

### 2.5.3 Shared DASD Considerations

In nonshared DASD environments, VCC is normally allowed to process all DASD volumes and catalogs. It is run once a day at a user-specified time to collect the data that will be used to manage and account for DASD space. In shared DASD environments, however, VCC must be handled in a different way to account for the following:

- Duplicate DASD volume serial numbers on each CPU that can be accessed by the same unit address, but are physically different devices (for example, the IPL volume for each CPU). Furthermore, data set names may be duplicated within each volume such as SYS1.NUCLEUS, SYS1.LPALIB, or SYS1.MANA.
- Devices that are accessible from one CPU but not from another, when all other devices are shared.
- Catalogs that are accessible from one CPU but not from another, when all other devices are shared.
- VSAM data sets on nonshared volumes that are pointed to from catalogs on shared volumes, and vice versa.
- A master catalog from one CPU that is defined as a user catalog in another CPU's master catalog.
- Large online applications (such as IMS) that hold long-term device reserves that may interfere with VCC trying to process that device from another CPU.

These situations require a site with shared DASD to carefully plan the methodology that will be used to collect DASD space data. We recommend that you use the following procedure:

- Create a separate VCC job stream for each CPU within the complex. Each job should be tailored to collect data for those devices that are accessible only from that CPU (that is, nonshared DASD devices and catalogs). VCC will add the SYSID to each volume and data set name that is collected, making each name unique.

- Create a VCC job stream to process only those DASD volumes and catalogs that reside on shared devices. Catalogs on nonshared devices should be excluded. The CPU that is issuing the long-term reserves should be selected to run this VCC job. The long-term reserve devices can be determined by examining RMF reports for the time period during which VCC will be run:

  - the Direct Access Device Activity Report (the % DEV RESV field)

  - the Reserve Activity Report (a Monitor II display)

  If the preferred CPU is down, another CPU in the complex can be used; however, the SYSID will be different for this collection run. You may want to wait until the preferred CPU is fixed.

Failure to use EXCLUDEVOL techniques on "secondary" VCC runs will result in duplicate data collection. If duplicate data set names or duplicate volume serial numbers are encountered, the SYSID will make each data set name and volume serial number unique. Thus, shared DASD data sets that are reprocessed again by another VCC will be treated as different data sets and will be entered into the CA MICS Database twice. This would increase the size of the database and would lead to overcharging for DASD space. You can solve this with careful use of EXCLUDEVOL lists.

VSAM data sets on shared volumes that are pointed to from catalogs on nonshared volumes (and vice versa) cannot be collected. You must identify and handle these data sets separately.

Make sure that master catalogs are not processed twice. It is common to have a master catalog from one CPU defined as a user catalog in another CPU's master catalog. These connected master catalogs should be excluded.

VCC uses the VCCNTROL file to retain information between executions. Information such as timestamps, and device and data collection statistics are recorded in the VCCNTROL file. This information is then used to adjust VCC processing according to your particular site. Some of this information is passed to the CA MICS Space Analyzer Option in order to properly compute the DURATION, a key element for accounting.

It is vital that each VCC run associated with a given SYSID DASD configuration be controlled by its own VCCNTROL file. VCC uses both ENQUEUE and RESERVE processing to ensure data integrity of the VCCNTROL file. The ENQUEUE or RESERVE of the VCCNTROL file is of short duration.

For a shared DASD environment, the VCCNTROL file cannot be shared between a VCC job to collect VSAM, non-VSAM, or DFHSM data, and the job to collect file system data.

ADDITIONAL CONSIDERATIONS FOR FILE SYSTEMS:

For HFS scan jobs collecting information on file systems which are shared between z/OS images, it is vital that each HFS scan job executing on separate systems do not share the same VCCNTROL file.

This situation requires a site with file systems shared between two or more z/OS images to carefully plan the methodology that is used to collect file systems data. CA recommends that you use the following procedure:

- Create a unique HFS scan job stream for each CPU within the complex. Each job should be tailored to collect data for those file systems that are accessible only from that CPU. HFS scan adds the SYSID to each volume and file system that is collected, making each name unique.

- When running HFS scan jobs in multiple images collecting information on file systems, care must be taken to exclude file systems that are shared among the multiple images to avoid duplicate data and over charging. This can be accomplished by coding the EXCLUDEDSN parameter to exclude the shared file systems that are already scanned by another HFS scan job.

  For file systems, using EXCLUDEVOL to avoid scanning of shared file systems that reside on shared DASD would result in dropping File Systems that are local to the system and should be scanned.

## 2.5.4 DASD Reserve Considerations

If SERIALIZE=N is specified as a run-time parameter, or is defaulted by the omission of the keyword SERIALIZE, then VCC will not ENQ on the SYSVTOC resource and will not RESERVE the volume as the VTOC is being read. When running VCC with SERIALIZE=N, there is a small possibility that the VTOC might be updated by SCRATCH, ALLOCATE, EXTEND, RENAME, or PARTIAL-RELEASE processes; this chance is very small because VCC reads an entire track of DSCBs with one I/O operation. Many large enterprises run the production VCC scanning job with SERIALIZE=N to eliminate the possibility of ENQ/RESERVE problems.

If absolute data integrity is necessary, then SERIALIZE=Y must be specified to cause VCC to issue an ENQ on SYSVTOC which results in a RESERVE on the volume. This serialization is released as soon as "logical end-of-VTOC" is reached; normally, this ranges from less than a second to a few seconds depending on many variables.

The remainder of this section describes a series of supplemental parameters that moderate the effect of SERIALIZE=Y.

- The BYPASSRES option allows the specification of DASD volume serial numbers that will be processed without being reserved first. To reduce the amount of keying required, generic values (prefixes) may be specified.

- The RESRETRY option allows you to specify the number of times VCC should attempt to reserve a volume. Each attempt actually consists of three reserves, issued one second apart. If the volume still cannot be reserved and the RESRETRY limit has not been exceeded, the volume will be placed back on the processing queue and will be tried again later. When the RESRETRY limit is exceeded, the action taken depends on the value of the RESFAILOK option.

- The RESFAILOK option allows you to specify the action to be taken when a volume cannot be reserved. Two options are available: the volume can be processed without being reserved (message VCCVT21W will be issued), or processing for the volume can be terminated (message VCCVT22W will be issued).

Section 3.1.1.2 of this guide provides details about these options.

### 2.5.5 Multi-Image Manager (MIM) Considerations

The Multi-Image Manager (MIM) product from CA can intercept hardware RESERVEs and translate them into less restrictive ENQs on other MVS images in a shared environment. This is a normal function of MIM but the default processing of MIM is to NOT perform this function when the major name associated with the serialization request is SYSVTOC.

Appendix A in the MIM Integrity System Guide describes why SYSVTOC is normally omitted from the QNAME list (MIM will not change the RESERVE into a global ENQ). The reasoning is that programs needing to serialize on SYSVTOC do so only for a brief period (only for few seconds); thus it is better to avoid the overhead of propagating ENQs to all connected system images.

VCC conforms to these guidelines in MIM. Under normal circumstances VCC processing is sufficiently fast that the SYSVTOC resource is only RESERVEd for a few seconds. Thus, the default setup of MIM is good for VCC when SERIALIZE=Y is specified (remember that SERIALIZE=N will not ENQ on SYSVTOC, therefore RESERVEs are never an issue).

Section 3.1.1.2 of this guide provides details about these options.

Note that if you do choose to override the default for MIM and add SYSVTOC to the MIM control file, you must also add several other major names to the MIM control file:

- SYSZVVDS
- SYSZRACF
- SYSIGGV2
- ARCBACV (if DFHSM is operational)
- ARCMIGV (if DFHSM is operational)
- DSPURI01 (if IMS is operational)

The above list is supplied as a warning for the considerations regarding SYSVTOC serialization imposed by VCC in a MIM environment. Always see the MIM documentation for the latest information on other considerations related to SYSVTOC.

# Chapter 3: USAGE GUIDELINES

This chapter describes how to use VCC.

This section contains the following topics:



## 3.1 Setting Up VCC for Operation

There are separate processes for both standard DASD space collection and file systems space collection. This was done to provide greater flexibility. For example, standard DASD should be collected on a minimum daily basis, while file systems collection may be performed for special studies only.

Once VCC is installed according to the detailed installation checklist in Chapter 4, the following data sets should have been allocated:

- The VCC control data set (described in the following sections as well as in Chapter 5).
- The VCC recording data set. This data set should be created only if SMF is NOT the VCC recording medium.

The sections that follow describe how to:

- Allocate and build the runtime parameter data set that tailors VCC to suit your needs.
- Create the JCL stream that you will be using to run VCC.

## 3.1.1 Selecting the Runtime Parameters

VCC offers a set of parameters that can be used to tailor VCC to your reporting needs. Parameters also have been provided to tune internal VCC constants that deal with VCC operating speed and the consumption of virtual storage.

These parameters apply to both standard VCC collection and HFS collection. The parameter descriptions are supplied in Section 3.1.1.2. To the right of each parameter name, it is specified whether the parameter is for standard VCC collection, HFS collection, or both.

The choice of parameters depends largely on your research during the VCC planning phase (see Chapter 2). The parameter data set can be a member of a partitioned, sequential, or "DD *" data set. The data set must be fixed blocked, physical sequential, with a logical record length of 80. This data set is referenced by the VCCPARMS DD statement (see Section 3.1.2). If the VCCPARMS DD statement is absent, then VCC will run using parameter defaults.

The sections that follow cover these runtime parameter topics:

1 - Runtime Parameter Features
2 - Runtime Parameter Specification

### 3.1.1.1 Runtime Parameter Features

The VCC runtime parameters can be broken down into the following functional areas:

- Scan limiting parameters
- Recording parameters
- VCC internal function parameters
- VCC report generation parameter

Scan Limiting Parameters

Both the standard VCC collection and HFS collection processes enable you to limit the scan to certain DASD volumes, VSAM or ICF catalogs, and other data sets. This is accomplished through the use of INCLUDE and EXCLUDE parameter keywords. Both specifications have been provided for ease of use, each being mutually exclusive of the other. For example, if you have 80 DASD volumes and wish to collect space utilization data for all except five system volumes, it is easier to build an EXCLUDE list specifying the five system volumes to exclude, rather than building an INCLUDE list specifying the 75 DASD volumes to be monitored by VCC.

### 3.1.1.2 Runtime Parameter Specification

VCC parameter specification follows much the same rules that govern most 80-byte card parameter formats. Every attempt has been made to simplify the format of the run-time parameters. The rules for the VCC parameter set are the following:

- Use columns 1 through 72 inclusive to enter VCC parameters.
- Begin parameters in any of the above columns of a card image.
- You cannot continue parameters that do not contain a list of values to another card.
- A parameter list is a series of values enclosed in parentheses and separated by commas. Nested parentheses are not allowed. You can continue a parameter list onto another card image by specifying a comma followed by a blank. The only limit to the number of continuation cards for a parameter is the maximum number of values that are allowed for the parameter.
- Specify all data set names, volume serial numbers, and device addresses generically by coding an asterisk (*) as the terminator for the value.
- Comment statements are recognized by an asterisk (*) in column 1 of a card image.
- Specify comments following each card image parameter statement or list value as long as there is at least one blank following the parameter specification.
- The VCC run-time parameter formats are divided into the following categories:
  - Parameter keywords with one operand.
  - Parameter keywords with a list of operands enclosed in parentheses. The parentheses must be coded, even if only one operand is entered.
- You must separate an operand from the parameter keyword specification by an equal sign (=) with no intervening blanks.

VCC Run-time Parameters Summary Table

| Parameter | Specification | Default | Notes |
|---|---|---|---|
| BYPASSBKUP | Y or N | N | 4,5 |
| BYPASSRES | (volser,...,volser) | | 4,5 |
| CATCHECK | Y or N | N | 3 |

| | | | |
|---|---|---|---|
| DATAINFO | Y or N | Y | 1 |
| DATALEVEL | (ALL,HFSDIR,TEMP) | ALL | 1,6 |
| DIAGDUMP | | | 6,8,9 |
| EXCLUDECAT | (dsn,dsn,...,dsn.*) | | 3 |
| EXCLUDEDEV | (dev,dev,...,dev) | | 3,H |
| EXCLUDEDSN | (dsn,dsn,...,dsn.*) | | 3 |
| EXCLUDEDV4 | (dev,dev,...,dev) | | 3,H |
| EXCLUDEPATH | (path,path,...path) | | 3,6 |
| EXCLUDEVOL | (volser,...,volser) | | 3 |
| EXCLUDVVDS | (dsn,dsn,dsn,dsn.*) | | 3 |
| EXTENDEDPDSE | (volser,...,volser) | | 1,2 |
| FXTHRESHOLD | (trk1,trk2,...,trk7) | | 3 |
| HFSERR | numeric value (1-16) | 0 | 4,6 |
| HFSDIRBUFFER | numeric value (8-256) | 16 | 4,6 |
| HFSDIRLEVEL | numeric value (1-512) | 32 | 4,6 |
| HFSSUBT | numeric value (1-16) | CPUs*3 | 4,6 |
| HSM | Y or N | N | 1 |
| HSMBACKUP | Y or N | Y | 1 |
| HSMMIGRATE | Y or N | Y | 1 |
| HSMSMF | Y or N | N | 1 |
| HSMSMFID | numeric value | 197 | 1 |
| INCLUDECAT | (dsn,dsn,...,dsn .*) | | 3 |
| INCLUDEDEV | (dev,dev,...,dev) | | 3,H |
| INCLUDEDSN | (dsn,dsn,...,dsn.*) | | 3 |
| INCLUDEDV4 | (dev,dev,...,dev) | | 3,H |
| INCLUDEPATH | (path,path,...path) | | 3,6 |
| INCLUDEVOL | (volser,...,volser) | | 3 |
| INCLUDVVDS | (dsn,dsn,dsn,dsn.*) | | 3 |
| LINECOUNT | numeric value (30-999) | 60 | 5 |
| NUMWTRBUF | numeric value (5-200) | 25 | 4,5 |
| OPERMSGS | Y or N | Y | 5 |
| REMOVEMNT | Y or N | N | 3 |
| RESERVEMNT | Y or N | N | 3 |
| RESFAILOK | Y or N | Y | 4,5 |
| RESRETRY | numeric value (1-250) | 10 | 4,5 |
| SERIALIZE | Y or N | N | 4,5 |
| SMF | Y or N | N | 1 |
| SMFID | numeric value | 198 | 1 |
| SPACEUSEDEQALLOC | Y or N | Y | 1,2,3 |
| SREPORT | Y or N | Y | 2 |
| STATUS | | | 6,8 |

| Parameter | Specification | Default | Notes |
|---|---|---|---|
| TEMPDSN | Y or N | N | 1 |
| TEST | Y or N | N | 4,9 |
| TRACE | Y or N | N | 4,5,9 |
| UNCATPDSE | Y or N | N | 4,5 |
| USEREXIT | Y, N or module_name | N | 6,7 |
| USERIDZERO | user_id_name | BPXROOT | 6 |
| VSAMERR | numeric value (1-16) | 0 | 4 |
| VSAMSUBT | numeric value (1-16) | 4 | 4 |
| VTOCERR | numeric value (1-16) | 0 | 4 |
| VTOCSUBT | numeric value (1-16) | 8 | 4 |
| WTO | Y or N | Y | 5 |

Notes Legend:

1 - Keyword affects data recording
2 - Keyword affects reporting
3 - Keyword affects scan processing
4 - Keyword affects performance
5 - Keyword modifies operational characteristics
6 - Keyword applies only to USS file system job (VCCRUNH)
7 - Module_name applies only to USS file system job (VCCRUNH)
8 - Parameter is supplied via z/OS Modify Operator command
9 - Technical Support use only
H - Keyword is not supported in file system job (VCCRUNH)

Parameter Detailed Descriptions (in alphabetical order)

Note: In the descriptions below, "standard" refers to the VCC collector, and "HFS" refers to the USS file system collector.

BYPASSBKUP (standard and HFS) - When specified as Y, VCC will bypass the check of the backup flag in the control file. This flag is set if a previous VCC run failed to perform a successful backup. All future VCC runs will fail until this flag is cleared. To skip checking this flag, specify BYPASSBKUP=Y.

Default: BYPASSBKUP=N

BYPASSRES (standard) - Specifies the volume serial numbers of volumes for which a hardware RESERVE should not be

issued before processing the VTOC. Consider specifying volumes that have a high level of RESERVE activity or tend to conflict with VCC and cause processing delays. Try to limit the size of this list, because the contents of a VTOC can change during processing if a RESERVE has not been issued. Specify a list of volume serial numbers. The list can contain generic specifications to reduce the amount of keying involved. A maximum of 2000 volume serial numbers may be specified in this list.

Example: BYPASSRES=(SYSRES,PAG*,TSOWK*)

CATCHECK (standard) - When specified as Y, VCC issues a LOCATE SVC for each VTOC data set that passes all filter criteria. The catalog status of the data set is recorded in a flag byte in the non-VSAM segment of the VCC record.

WARNING: CATCHECK processing increases the amount of time it takes VCC to complete the processing of a given VTOC. Therefore, coding CATCHECK=Y forces VCC to set SERIALIZE=N.

The CATCHECK feature provides an easy way to identify VTOC data sets that are not cataloged or that are cataloged to a different volume in preparation for SMS migration, specifically volume conversion. Use CATCHECK=Y infrequently since the LOCATE SVC function adds significant overhead. For example, CPU time will more than double.

One suggestion is to establish two different members of a VCCPARMS data set, and have a job scheduling package or production control system use a "standard" set of parameters for each VCC run, except for the Sunday of each week. On Sunday, CATCHECK=Y would lengthen the elapsed time for VCC but provide a cycle in the database that could be examined for uncataloged "problem" data sets.

The CA MICS StorageMate Option contains a series of standard reports that exploit the CATCHECK feature by identifying problem data sets that will inhibit SMS migration.

```
            Default:  CATCHECK=N


DATAINFO    (standard) - Specifies whether VCC records data
            set information or writes only volume information
            to the VCCDATA file (or SMF).  DATAINFO=N produces
            the fastest possible scan with one record produced
            for each volume scanned.  Perhaps one scan per day
            is run with DATAINFO=Y to collect information for
            accounting applications and other scans are run
            every hour with DATAINFO=N for storage
            applications.

            Default:  DATAINFO=Y


DATALEVEL (HFS) - Specifies the level of information that VCC
            will record. The operands for this keyword are the
            following:

            ALL     - VCC will record all levels of
                      information.
            HFSDIR  - VCC will record file system and
                      directory level information for all
                      Hierarchical File Systems.
            TEMP    - VCC will record files stored on the
                      z/OS UNIX Temporary File System (TFS).

            Default: DATALEVEL(ALL)
            Example: DATALEVEL(TEMP,HFSDIR)


DIAGDUMP (HFS) - Facilitates problem solving by enabling VCC
                 to produce diagnostic information.  To be
                 used at the direction of Technical Support.
                 For assistance, contact Technical Support at
                 http://ca.com/support.


EXCLUDECAT/INCLUDECAT (standard) - Specifies the names of the
            catalogs that are not eligible or that are
            eligible for processing by VCC.  INCLUDECAT or
            EXCLUDECAT is specified as a list of data set
            names.  Up to 200 data set names may be included
            in the list.  You can reduce the amount of keying
            involved by listing generic specifications.
            INCLUDECAT and EXCLUDECAT are mutually exclusive.
            By default, VCC scans all catalogs it can find
```

```
            connected to the master catalog.

            Example:    EXCLUDECAT=(SYS1.XCAT.*,SYS1.USERCAT1,
                        SYS1.USERCAT2)

EXCLUDEDEV/INCLUDEDEV (standard) - Specifies three-digit
            device addresses that are not eligible or that are
            eligible for processing by VCC.  Specify
            INCLUDEDEV or EXCLUDEDEV as a list of device
            addresses.  Up to 2000 device addresses may be
            included in the list.  You can reduce the amount
            of keying involved by using generic
            specifications. Only one of the following
            parameters may be used for a VCC scan:
            INCLUDEDEV, EXCLUDEDEV, INCLUDEDV4, or EXCLUDEDV4.
            By default VCC scans all DASD volumes that are
            online, resident, and ready.  If you also code the
            INCLUDEVOL/EXCLUDEVOL parameter, both the volume
            serial number and device address tests must be met
            or VCC will not select the volume.

            Example:  EXCLUDEDEV=(480,3B1,8FF,52*,54*,900)


EXCLUDEDSN/INCLUDEDSN (standard and HFS) - Specifies the data
            sets that are not eligible or that are eligible
            for processing by VCC.  INCLUDEDSN or EXCLUDEDSN
            is specified as a list of data set names.  Up to
            200 data set names may be specified in the list.
            This list can contain generic specifications to
            reduce the amount of keying involved.  INCLUDEDSN
            and EXCLUDEDSN are mutually exclusive.  By
            default, VCC records DASD space utilization data
            for all data sets it encounters during the VTOC
            and catalog scans.

            Example:  EXCLUDEDSN=(SYS1.*,SYS2.*,SYS3.*)


EXCLUDEDV4/INCLUDEDV4 (standard) - Specifies four-digit
            device addresses that are not eligible or that are
            eligible for processing by VCC.  Specify
            INCLUDEDV4 or EXCLUDEDV4 as a list of device
            addresses.  Up to 2000 device addresses may be
            included in the list.  You can reduce the amount
            of keying involved by using generic
            specifications.  Only one of the following
            parameters may be used for a VCC scan:
            INCLUDEDV4, EXCLUDEDV4, INCLUDEDEV, or EXCLUDEDEV.
```

```
                By default, VCC scans all DASD volumes that are
                online, resident, and ready.  If you also code the
                INCLUDEVOL/EXCLUDEVOL parameter, both the volume
                serial number and device address tests must be met
                or VCC will not select the volume.

                Example:  EXCLUDEDV4=(04*,13B1,18FF,252*,254*,29*)


EXCLUDEPATH/INCLUDEPATH (HFS) - Specifies the HFS paths that
           are not eligible or that are eligible for
           processing by VCC.  INCLUDEPATH or EXCLUDEPATH is
           specified as a list of HFS paths. Up to 200 paths
           may be included in the list. This list can contain
           generic specifications to reduce the amount of
           keying involved. INCLUDEPATH and EXCLUDEPATH are
           mutually exclusive. By default, VCC scans all HFS
           paths it can find.  If you also code any other
           INCLUDE or EXCLUDE parameter, all tests must be
           met for the HFS path to be scanned.


EXCLUDEVOL/INCLUDEVOL (standard and HFS) - Specifies the
           volume serial numbers that are not eligible or
           that are eligible for processing by VCC.
           INCLUDEVOL or EXCLUDEVOL is specified as a list of
           volume serial numbers.  Up to 2000 volume serial
           numbers may be included in the list.  This list
           can contain generic specifications to reduce the
           amount of keying involved.  INCLUDEVOL and
           EXCLUDEVOL are mutually exclusive.  By default,
           VCC scans all DASD volumes that are online,
           resident, and ready.  If you also code the
           INCLUDEDEV/EXCLUDEDEV parameter, both the volume
           serial number and device address tests must be met
           or VCC will not select the volume.

           Example:  INCLUDEVOL=(IPLA01,TSO*,SCR001)


EXCLUDVVDS/INCLUDVVDS (standard) - Specifies the names of the
           VSAM volume data sets that are not eligible or
           that are eligible for processing by VCC.
           INCLUDVVDS or EXCLUDVVDS is specified as a list of
           data set names.  Up to 200 data set names may be
           included in the list.  This list can contain
           generic specifications to reduce the amount of
           keying involved.  Also, because all VVDS names
           start with the characters 'SYS1.VVDS.V', you can
```

use a dollar sign ('$') in this parameter to replace these characters. Thus, SYS1.VVDS.VSYSRES and $SYSRES are logically identical. INCLUDVVDS and EXCLUDVVDS are mutually exclusive. By default, VCC will scan all VVDSs it can find. This parameter is useful in limiting the scope of a VCC scan for problem-solving purposes, eliminating bad VVDSs from scanning until they are fixed.

Example: EXCLUDVVDS=(SYS1.VVDS.VIPL*,$PAGE*)

EXTENDEDPDSE (standard) - Specifies how collection of the extended PDS/E attribute data is to be handled. This parameter functions at either the global or local level. At the global level the parameter controls whether or not the extended PDS/E attribute data is collected for all DASD volumes that are to be scanned. At the local level the parameter can be used to specify specific volumes or groups of volumes that will or will not have the extended PDS/E attribute data collected.

EXTENDEDPDSE or NOEXTENDEDPDSE can be specified as a list of volume serial numbers. Up to 2000 volume serial numbers may be included in the list. This list can contain generic specifications to reduce the amount of keying involved. EXTENDEDPDSE and NOEXTENDEDPDSE are mutually exclusive. By default, VCC collects extended PDS/E attribute data for all DASD volumes that are online, resident, and ready.

Examples:

1. EXTENDEDPDSE

   Specifies collection of extended PDS/E attribute data for all scanned DASD volumes.

2. NOEXTENDEDPDSE

   Specifies that collection of extended PDS/E attribute data will NOT be performed for any scanned DASD volumes.

3. EXTENDEDPDSE(ABC001,XYZ*)

```
                    Specifies collection of extended PDS/E
                    attribute data for DASD volume 'ABC001' and
                    all DASD volumes where the volume serial
                    numbers begin with the characters 'XYZ'.

                4.  NOEXTENDEDPDSE(ABC001,XYZ*)

                    Specifies that collection of extended PDS/E
                    attribute data will NOT be performed for
                    DASD volume 'ABC001' or any DASD volumes
                    where the volume serial numbers begin with
                    the characters 'XYZ'.


FXTHRESHOLD (standard) - Specifies the thresholds to be used
           in collecting the free-extent buckets count.
           These counts are populated into the VOASPF01
           through VOASPF08 data elements.  This parameter
           allows the user to specify the threshold, in
           tracks, for the first seven free-extent buckets.
           The eighth free-extent bucket threshold is always
           set to high value in order to count any extent
           sizes not contained within the first seven defined
           buckets.

           Example:

             FXTHRESHOLD=(100,200,300,400,600,800,1000)

           The values shown in the above example are the
           default values used in case a threshold statement
           is not supplied.  The default values will yield
           the following counts in the free extent buckets:

             VOASPF01 -    1 to  100 track free extents
             VOASPF02 -  101 to  200 track free extents
             VOASPF03 -  201 to  300 track free extents
             VOASPF04 -  301 to  400 track free extents
             VOASPF05 -  401 to  600 track free extents
             VOASPF06 -  601 to  800 track free extents
             VOASPF07 -  801 to 1000 track free extents
             VOASPF08 - 1001 and over track free extents


HFSERR     (HFS) - Defines the maximum number of subtasks
           that VCC uses to scan file systems that are
           allowed to terminate with a "critical" error
           without affecting the entire run (other subtasks).
```

Specify this parameter as a number between 0 and 16. The default is 0. The specified value should never exceed the number of tasks specified in HFSSUBT keyword minus 1.

Example: HFSERR=3
HFSSUBT=4

See Section 5.3, Abnormal Termination Processing, for a discussion of considerations when choosing this value.

HFSDIRBUFFER (HFS) - Defines the size of individual buffers that are obtained to scan file systems. This value is specified in units of 1024 bytes. (For example, HFSDIRBUFFER=4 equals 4*1024 of 4096 bytes per buffer.) This parameter functions in conjunction with the HFSDIRLEVEL parameter in that there is one buffer obtained for every directory level specified by the HFSDIRLEVEL parameter. A larger size can yield better performance, but it will also cause more virtual storage to be obtained during scanning.

Default: HFSDIRBUFFER=16

HFSDIRLEVEL (HFS) - Defines the number of file system (HFS) directory levels that can be scanned. Each HFS path name is comprised of one or more nodes. (For example, /aaaa/bbbb/cccc/ contains three nodes or directory levels.) Complex path names typically consist of numerous nodes requiring a higher value for the parameter. This value must be set to a value that is as large as or larger than the maximum number of directory nodes that will be processed.

Default: HFSDIRLEVEL=32

This parameter is no longer of any relevant importance. The HFS scan logic has been improved to dynamically adapt to increases in the number of directory levels. This eliminates the problem of potential reruns caused by exceeding the number of directory levels specified by the HFSDIRLEVEL parameter.

```
HFSSUBT     (HFS) - Defines the number of subtasks that VCC
            uses to scan Hierarchical File Systems that are
            mounted and reside on z/OS.  Specify this
            parameter as a number between 0 and 16.  The
            specified value should never exceed the number of
            devices being processed during a particular scan
            execution.

            Default:  HFSSUBT=(number_of_processors * 3)

            See the VTOCSUBT parameter below for a discussion
            of considerations when choosing this value.


HSM         (standard) - Specifies whether the VCC-HSM
            interface should be attached to collect data about
            data sets under the management of DFHSM.

            This interface runs as a separate subtask creating
            data records in the VCCHSM data set (or SMF) and
            creates its own separate summary report on the
            VCCHSMSG file.

            This parameter value has the effect of
            establishing defaults for other parameters related
            to HSM processing.  For example, HSMBACKUP=Y has
            no effect unless HSM=Y is in effect. If HSM=Y is
            in effect, HSMBACKUP=Y is the default.

            Default:    HSM=N


HSMBACKUP   (standard) - Specifies whether the VCC-HSM
            interface should collect data from the HSM Backup
            Control Data Set (BCDS).  There are potentially
            many records that will be created.  If your
            application does not anticipate using this data,
            you can save elapsed time and space on the VCCHSM
            data set and in the database by disabling the
            collection of data about backup versions.

            The default is Y whenever HSM=Y is coded.

            Default:    HSMBACKUP=Y


HSMMIGRATE (standard) - Specifies whether the VCC-HSM
            interface collects data from the HSM Migration
```

Control Data Set (MCDS). With this option, the VCC-HSM interface can track data sets across the entire hierarchy of storage managed by HSM.

The default is Y whenever HSM=Y is coded.

Default: HSMMIGRATE=Y

HSMSMF (standard) - Specifies whether the VCC-HSM interface should write the output data records to SMF or to the VCCHSM data set.

We recommend you write the HSM records to the VCCHSM data set rather than log them directly to SMF. This is because of the relatively high volume of records this interface creates and the associated probability of overflowing the SMF buffer.

If you want to consolidate the data on SMF, you could consider appending the VCCHSM data set to the dumped SMF data set as an alternative to direct SMF logging.

Default: HSMSMF=N

HSMSMFID (standard) - Specifies the SMF record number the VCC-HSM interface uses in writing out data records.

You should only change this number if you are actually going to log data to SMF via the HSMSMF=Y parameter above. If you change this from the default value, you will need to coordinate a LOCAL MODIFICATION to the sharedprefix.MICS.GENLIB member named HSMGENIN.

If remote data centers have HSM data to be collected and processed by CA MICS at a central location, the HSMSMFID value chosen must be the same by the time it is processed by a central CA MICS. If you cannot synchronize this, a copy utility could change the byte in the record to the common value coded in the HSMGENIN member of sharedprefix.MICS.GENLIB.

If you want to consolidate the data on SMF, you

could consider appending the VCCHSM data set to the dumped SMF data set.

Default: HSMSMFID=197

LINECOUNT (standard and HFS) - Specifies the maximum number of lines per page for the message log. While writing messages to the message log, VCC will skip to a new page and generate the heading lines each time this value is reached. The value specified for LINECOUNT is also used for VCC Statistics Reports as well.

Default: LINECOUNT=62

NUMWTRBUF (standard and HFS) - Specifies the number of buffers that VCC should obtain for the output writer subtask. The various VCC subtasks use these buffers to store output records while the VCC writer subtask is busy writing information to the VCCDATA output file or to SMF. Specifying too small a value will cause delays while the subtask waits for a buffer to become available. Specifying too large a value will increase both the region size and CPU usage of VCC. Consider using a larger value if you are writing scan data to a data set, rather than to SMF. Specify a number between 10 and 100.

Default: NUMWTRBUF=25

OPERMSGS (standard and HFS) - Specifies whether or not VCC should notify operations if unusual or catastrophic situations occur during a VCC run. VCC will issue Write-to-Operator messages (WTOs) to the master console in these situations.

Default: WTO=Y

REMOVEMNT (standard and HFS) - Specifies whether or not DASD devices that are neither permanently resident nor reserved are to be scanned by VCC. Removable volumes can be dismounted either at the end of the job in which they were last used or when the unit on which the volume is mounted is needed for

another volume.

Default: REMOVEMNT=N

RESERVEMNT (standard and HFS) - Specifies whether or not DASD devices that are mounted with the PRIVATE, RESERVED attributes are to be scanned by VCC.

Default: RESERVEMNT=N

RESFAILOK (standard) - Specifies the action that VCC should take when it cannot RESERVE a volume that it is attempting to process. You control the number of times that VCC will attempt the RESERVE by the RESRETRY parameter. Specify RESFAILOK=Y (Yes) if you want VCC to continue processing the volume without reserving it. Use this option with caution, as VTOCs that are not reserved can have their data changed while being processed by VCC. Specify RESFAILOK=N (No) if you want VCC to bypass the volume and not process it.

Default: RESFAILOK=Y

RESRETRY (standard) - Specifies the number of times that VCC attempts to RESERVE a volume before taking the action specified by the RESFAILOK parameter. A retry consists of three RESERVEs, one second apart. If VCC cannot reserve the volume in these three attempts and the RESRETRY limit has not been exceeded, VCC places the volume back on the queue and tries later with another subtask. Thus, specifying RESRETRY=4 causes up to 12 RESERVEs (3 times 4) to be issued for a volume before taking the RESFAILOK action. Use this option with caution, as VTOCs that are not reserved can have their data changed while being processed by VCC. Specify a number between 1 and 250.

Default: RESRETRY=10

SERIALIZE (standard) - Controls issuing of a cross-system ENQ on the SYSVTOC resource for a given volume. This type of ENQ results in a hardware RESERVE being issued for the entire device until the

SYSVTOC resource is released (DEQueued). This option preserves the integrity of reading a VTOC to ensure that VCC can obtain the correct data set information. Specifying a Y causes an ENQ/RESERVE to be issued before the start of each VTOC scan. Specifying an N bypasses the issuing of the ENQ/RESERVE for ALL VTOCs during the current execution. There is a small possibility that not reserving the VTOC during the scan can cause data integrity problems. Reserving the VTOC can cause operational problems with other jobs and online systems, ESPECIALLY when VCC is run during a prime shift or a busy period.

Note: We recommend that you always use SERIALIZE=N to avoid operational bottlenecks due to ENQ waits. Never use SERIALIZE=Y during peak load. The only time you should use SERIALIZE=Y is when absolute data integrity is necessary (for example, when you are utilizing Accounting and Chargeback data) and only then when it is an off-peak time.

You should consider BYPASSRES lists for "system" type volumes such as PAGE, SPOOL, and (DFHSM) ML1 volumes.

SMF (standard and HFS) - Specifies the recording medium VCC uses to record DASD space utilization data. Y (yes) tells VCC to record to SMF. N (no) tells VCC to record to a data set.

Default: SMF=N

Note: If you code SMF=Y, make sure z/OS is prepared to write the proper SMF record as coded by SMFID (below). To verify this, check SYS1.PARMLIB(SMFPRMxx) as discussed in the appropriate IBM documentation.

SMFID (standard and HFS) - Specifies the SMF record number that VCC places in each record it writes. You can specify SMFID as a number between 128 and 255 inclusive. If the default conflicts with an SMF record number being produced by another product, change the record number VCC produces with this parameter.

Note: CA MICS must also know about any changes to this parameter.

If remote data centers have DASD data to be collected and processed by CA MICS at a central location, the SMFID value chosen must be the same by the time it is processed by a central CA MICS. If this cannot be synchronized, a copy utility could change the byte in the record to the common value coded in the VCAGENIN member of sharedprefix.MICS.GENLIB.

Default: SMFID=198

SPACEUSEDEQALLOC/NOSPACEUSEDEQALLOC (standard) - This parameter, when specified, causes the VTOC scan tasks to set the value of the space-used field in the collection record to the space-allocated value for PDS/E and HFS data sets. The default for this parameter is on and may only be turned off (NOSPACEUSEDEQALLOC) when used in conjunction with the specification of the DATAINFO=N parameter, which produces only the volume total records for a DASD volume. See also the PDSEUSED/NOPDSEUS parameter in the CA MICS Space Analyzer Option guide for additional information regarding the interaction of the parameters.

Default: SPACEUSEDEQALLOC

SREPORT (standard) - Determines the sort order of the VCCSTATS report. The default shows the volumes processed in sequence by volume serial number. Coding SREPORT=N overrides this to list volumes by device address.

Default: SREPORT=Y

STATUS (HFS) - This parameter is issued via the z/OS Modify Operator command (F jobname,STATUS) to produce status information regarding VCC processing. It is only available to the batch job that is used to obtain z/OS UNIX HFS information.

TEMPDSN (standard) - Provides for the automatic

recognition and exclusion of system-generated temporary data sets from the VCC inventory. The space occupied by a temporary data set is still counted toward the totals kept for the VCAVOA file, but TEMPDSN=N prevents VCC from writing a record to the VCCDATA file for the temporary data set itself.

You should remove any EXCLUDEDSN=(SYS94*) statements you may have in your VCCPARMS.

Default: TEMPDSN=N

TEST (standard and HFS) - Facilitates problem solving by enabling VCC to produce diagnostic information. The Y (yes) specification enables this facility, which may result in voluminous output. Use the INCLUDE/EXCLUDE run-time parameters described in this section to limit the amount of output.

This parameter is intended to be used at the direction of Technical Support. For assistance, contact Technical Support at http://ca.com/support.

Default: TEST=N

TRACE (standard and HFS) - Facilitates problem solving by enabling VCC to produce diagnostic information. Enabling this facility produces voluminous output. Use the INCLUDE/EXCLUDE run-time parameters described in this section to limit the amount of output.

This parameter is intended to be used at the direction of Technical Support. For assistance, contact Technical Support at http://ca.com/support.

Default: TRACE=N

UNCATPDSE (standard) - Specifies whether VCC is to collect extended attribute information for uncataloged PDS/E data sets. Normal VCC processing only collects extended attribute information for cataloged PDS/E data sets. Since PDS/E data sets

are not required to be cataloged, there could be a considerable number of data sets for which extended attribute information would not be collected. This information includes the actual space utilization within the PDS/E data sets.

One potential side effect of enabling this parameter is the possibility that there may be corrupted PDS/E data sets residing on DASD volumes. Certain corrupted PDS/E data sets may cause S0F4 abend conditions to appear while VCC is processing. These abends occur in a system service and cannot be intercepted by the VCC modules. The system service recovers from the abend and processing continues. The only loss is the extended attribute data for the corrupted PDS/E data set; otherwise, normal processing continues. Information about the corrupted data sets is written to the VTOC Scan Log and may be used to analyze the data sets in question. A possible solution is to delete the corrupted PDS/E data set. If there are a large number of these data sets, you may want to modify the VCC run JCL and change the DD statement for the SYSUDUMP output to the following:

```
//SYSUDUMP DD DUMMY
```

This will prevent the output queue from filling up as a result of the dumps produced by the S0F4 abends. Once the corrupted data sets have been identified and corrected or deleted, the normal SYSUDUMP specification can be returned.

Default: UNCATPDSE=N

USEREXIT (standard) - Specifies whether VCC invokes a user exit to perform user processing of Format 1 VTOC DSCBs and the VCC output record. See Chapter 5 for more information. For the batch job used to collect HFS information, this parameter allows specification of the module name of the user exit that performs additional processing to the VCC output records.

Default: USEREXIT=N or USEREXIT=NONE.

```
USERIDZERO (HFS) - Specifies a 1 to 8 character name value
           as a default user ID to be written to the output
           records during HFS scan processing. This
           parameter has no effect on the security access to
           HFS file structures required for the scan.

           Default:  USERIDZERO=BPXROOT


VSAMERR    (standard) - Defines the number of VSAM/ICF
           catalog scan subtasks that can terminate with a
           "critical" error without affecting the entire run
           (other subtasks).

           Specify this parameter as a number between 0 and
           31.  The default is 0.  The specified value should
           not exceed the number of tasks specified in
           VSAMSUBT keyword minus 1.

           Example:  VSAMERR=3
                     VSAMSUBT=4

           See Section 5.3, Abnormal Termination Processing,
           for a discussion of considerations when choosing
           this value.


VSAMSUBT   (standard) - Defines the number of subtasks that
           VCC uses to scan VSAM/ICF catalogs and VVDS data
           sets.  Specify this parameter as a number between
           0 and 32.  The value you specify should never
           exceed the number of devices being processed
           during a particular scan execution.

           Default:  VSAMSUBT=4

           See the VTOCSUBT parameter for a discussion of
           considerations when choosing this value.


VTOCERR    (standard) - Defines the number of VCCVTOC
           subtasks that can terminate with a "critical"
           error without affecting the entire run (for
           example, other subtasks).

           Specify this parameter as a number between 0 and
           31.  The default is 0.  The value you specify
           should not exceed the number of tasks specified in
           VTOCSUBT keyword minus 1.
```

Example: VTOCERR=2
VTOCSUBT=8

See Section 5.3, Abnormal Termination Processing, for a discussion of considerations when choosing this value.

VTOCSUBT (standard) - Defines the number of subtasks to be used to scan VTOCs. Specify this parameter as a number between 0 and 32. The specified value should never exceed the number of devices being processed during a particular scan execution.

Default: VTOCSUBT=8

VCC uses the values of the VTOCSUBT, VSAMSUBT, and HFSSUBT parameters to govern how fast it scans the DASD configuration. The main purpose is to make the most efficient use of I/O wait time by overlapping VCC I/O requests between subtasks. (While one subtask is waiting for I/O to complete, others can be processing their data.)

This subtask facility also provides a means by which VCC I/O can be distributed across the I/O configuration.

Use these parameters wisely, especially in the case of the VTOC scan, because a large amount of memory is obtained for I/O buffers and DSCB chain resolution queues by each VTOC scan subtask.

WTO (standard) - (Write To Operator) Specifies whether or not VCC should notify operations if unusual or catastrophic situations occur during a VCC run. VCC will issue WTOs to the master console in these situations. See Chapter 7, Return Codes and Messages, for message identifier VCCCN13I.

Default: WTO=Y

## 3.1.2 Customizing the VCC JCL

This section summarizes the Job Control Language (JCL) needed to run VCC. The following two members have been provided in the library sharedprefix.MICS.CNTL:

- VCCRUN - To run the Standard VCC Collection (see Section 3.1.2.1).
- VCCRUNH - To run the Hierarchical File System Collection (see Section 3.1.2.2).

After modifying the JCL, the CA MICS Space Collector Option is ready for operation. When the CA MICS Space Analyzer Option (VCA) is installed and configured to your requirements, the data generated by VCC may be fed to VCA via the CA MICS Daily job.

### 3.1.2.1 Sample JCL Member VCCRUN

The following JCL is used for collecting DASD VTOC, VSAM, HSM, and System Catalog information.

```
//jobname  JOB  (),'',
//         CLASS=A,
//         MSGCLASS=A,
//         NOTIFY=
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*                     - Sample JCL -                           *
//*              CA MICS DASD Space Collector (VCC)              *
//*   VTOC, VSAM, HSM, and System Catalog Data Collection        *
//*                                                              *
//* Notes:                                                       *
//*   1. VCC requires a minimum of 4096K to run.                 *
//*                                                              *
//*   2. If not included in the system linklist, the STEPLIB*
//*      must be an authorized library.                          *
//*                                                              *
//*   3. Use of this data set is optional. It is used for        *
//*      recording information from DFHSM.                       *
//*                                                              *
//*   4. Use of these data sets is optional.  They are used      *
//*      for collecting information from DFHSM. Specify the      *
//*      data set names for the MCDS and BCDS DFHSM data         *
//*      sets.  You may remove these DD statements allowing      *
//*      VCC to dynamically allocate them if DFHSM               *
//*      information is desired.                                 *
//*                                                              *
//*   5. If DFHSM information is to be collected and your        *
//*      data center is using a split MCDS or BCDS, you          *
//*      should code your DD statements as follows:              *
//*                                                              *
//*      For a split MCDS:                                       *
//*      //MCDS     DD   DISP=SHR,DSN=_____._______.MCDS1      *
//*      //MCDS2    DD   DISP=SHR,DSN=_____._______.MCDS2      *
//*      .                                                       *
//*      .                                                       *
//*      .                                                       *
//*                                                              *
//*      For a split BCDS:                                       *
//*      //BCDS     DD   DISP=SHR,DSN=_____._______.BCDS1      *
//*      //BCDS2    DD   DISP=SHR,DSN=_____._______.BCDS2      *
//*      .                                                       *
//*      .                                                       *
//*      .                                                       *
//*                                                              *
```

```
//*      Note that in both instances the first DD name is   *
//*      MCDS or BCDS, not MCDS1 or BCDS1.                  *
//*                                                         *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//*-----------------------------------------------------------*
//*STEP1 - Collect DASD VTOC, VSAM, HSM, System Catalog Data*
//*-----------------------------------------------------------*
//*
//STEP1    EXEC PGM=VCCNTRL,REGION=5000K     <== See Note-1
//*
//STEPLIB  DD   DISP=SHR,DSN=__________      <== See Note-2
//VCCSTATS DD   SYSOUT=*
//VCCVTMSG DD   SYSOUT=*
//VCCVSMSG DD   SYSOUT=*
//VCCHSMSG DD   SYSOUT=*
//VCCSNAP  DD   SYSOUT=*
//ARCSNAP  DD   SYSOUT=*
//*
//VCCNTROL DD   DISP=SHR,DSN=______.VCC.CONTROL
//*
//VCCDATA  DD   DSN=______.VCC.VCCDATA1,
//         DISP=OLD
//*
//*
//*
//VCCHSM   DD   DSN=______.VCC.HSMDATA,      <== See Note-3
//         DISP=OLD
//*
//*
//*
//MCDS    DD DISP=SHR,DSN=_____._______.MCDS  <== See Note-4,5
//**MCDS2 DD DISP=SHR,DSN=_____._______.MCDS2 <== See Note-5
//**MCDS3 DD DISP=SHR,DSN=_____._______.MCDS3 <== See Note-5
//BCDS    DD DISP=SHR,DSN=_____._______.BCDS  <== See Note-4,5
//**BCDS2 DD DISP=SHR,DSN=_____._______.BCDS2 <== See Note-5
//**BCDS3 DD DISP=SHR,DSN=_____._______.BCDS3 <== See Note-5
//*
//VCCPARMS DD   *
VSAMSUBT=8
VSAMERR=4
VTOCSUBT=8
VTOCERR=4
/*
//*
//*-----------------------------------------------------------*
//* STEP2  -  Copy collected data to SMF DAILY GDG          *
//*-----------------------------------------------------------*
//*
```

```
//STEP2    EXEC PGM=IEBGENER,COND=(4,LT,STEP1)
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=______.VCC.VCCDATA1,
//         DISP=OLD
//SYSUT2   DD   DSN=______.VCC.SMFBKUP(+1),
//         UNIT=____,
//         DISP=(,CATLG,DELETE),LABEL=(1,SL)
//SYSIN    DD   DUMMY
//*
//*-------------------------------------------------------------*
//* STEP3  -  Copy collected HSM data to SMF DAILY GDG      *
//*-------------------------------------------------------------*
//*
//STEP3    EXEC PGM=IEBGENER,COND=(4,LT,STEP1)
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=______.VCC.HSMDATA,
//         DISP=OLD
//SYSUT2   DD   DSN=______.VCC.HSMBKUP(+1),
//         UNIT=____,
//         DISP=(,CATLG,DELETE),LABEL=(1,SL)
//SYSIN    DD   DUMMY
//*
//*-------------------------------------------------------------*
//* STEP4  -  Flag VCC Data Copied                          *
//*-------------------------------------------------------------*
//*
//STEP4    EXEC PGM=VCCUPDT,REGION=512K,
//         COND=((4,LT,STEP1),(0,NE,STEP2),(0,NE,STEP3))
//STEPLIB  DD   DISP=SHR,DSN=__________      <== See Note-2
//*
//VCCNTROL DD   DISP=SHR,DSN=______.VCC.CONTROL
//*
```

VCCRUN JCL Detail Description

Jobcard

You can run VCC in any job class except a JES3 job class that has JOURNALING specified. You may need to specify a region size if your default size is not at least 2048 K.

Step 1

This step will execute VCC to collect DASD information.

STEPLIB - This statement is required only if the VCC

modules were not placed in an authorized library that is in the system linklist. Otherwise, it should point to the authorized library to which the VCC modules were moved during installation. (See Chapter 4, Installation).

VCCSTATS - Used for the VTOC scan message log, as well as for the VCC Run Status Report. If pre-allocated to a disk data set, DCB attributes should include RECFM=VBA,LRECL=133.

VCCVSMSG - Defines the DD statement that VCC uses for the catalog scan log. If pre-allocated to a disk data set, DCB attributes should include RECFM=VBA,LRECL=133.

VCCVTMSG - Defines the DD statement that VCC uses for the VTOC scan log. If pre-allocated to a disk data set, DCB attributes should include RECFM=VBA,LRECL=133.

VCCSNAP - VCC uses this DD statement to SNAP memory for the VCC TEST option. It is also used during abnormal termination of a subtask to produce a SNAP dump.

VCCPARMS - This DD statement is optional. If present, it defines the VCC parameter data set. This data set can be a member of a PDS, a physical sequential file, or a DD * and you can code the parameters as part of the JCL stream.

VCCDATA - This DD statement defines the data set that VCC will use as a recording medium if SMF is not going to be used. This data set must be variable blocked. The LRECL and BLKSIZE are controlled by VCC in order to ensure maximum density of data per track. At run time, VCC will override any values you code on the VCCDATA DD statement.

| Device | Parameters Enforced by VCC |
|---|---|
| 3390 | DCB=(BLKSIZE=27998,LRECL=27994,RECFM=VB) |
| 3380 | DCB=(BLKSIZE=23476,LRECL=23472,RECFM=VB) |
| other | DCB=(BLKSIZE=32760,LRECL=32756,RECFM=VB) |

VCCNTROL - Defines the VCC control file. You allocated it during VCC installation and it keeps track of the volumes processed and VCC processing timestamps.

The control file must be unique for each z/OS system that will run VCC. It must not be shared between z/OS images within a complex.

Data Sets Used by The VCC Interface to DFHSM (VCC-DFHSM)

VCCHSM - This DD statement defines the data set that VCC uses as a recording medium if SMF is not used. If you are writing HSM data directly to the VCCHSM data set, you should allocate a large data set. If you are using SMF as the recording source for HSM data, the file will be used as a work file and you need to allocate only a few tracks. If the data set is omitted, VCC will not process HSM data and will issue a return code 8 for this step.

| Device | Parameters Enforced by VCC |
|---|---|
| 3390 | DCB=(BLKSIZE=27998,LRECL=255,RECFM=VB) |
| 3380 | DCB=(BLKSIZE=23476,LRECL=255,RECFM=VB) |
| other | DCB=(BLKSIZE=32760,LRECL=255,RECFM=VB) |

VCCHSMSG - Defines the DD statement that VCC uses for the DFHSM Interface Report. If pre-allocated to a disk data set, DCB attributes should include RECFM=VBA,LRECL=137.

MCDS - Defines the DD statement that specifies the data set name of the DFHSM Migration Control Data Set. This is required if HSMMIGRATE=Y is coded or defaulted.

Note: Consult with your systems programmer or DFHSM administrator on the data set names for the MCDS and BCDS and decide which z/OS image in a shared DASD environment will be responsible for initiating a VCC inventory of DFHSM control data sets.

MCDS2, MCDS3, and so on - Define the DD statements that describe additional DFHSM Migration Control Data Sets, if the installation employs a split MCDS. Note that there is no such DD name as MCDS1.

BCDS - Defines the DD statement that specifies the data set name of the DFHSM Backup Control Data Set. This is required if HSMBACKUP=Y is coded or defaulted.

BCDS2, BCDS3, and so on - Define the DD statements that describe additional DFHSM Backup Control Data Sets, if the installation employs a split BCDS. Note that there is no such DD name as BCDS1.

ARCSNAP - Defines a SYSOUT data set that the DFHSM ARCUTIL program opens when LINKed by the VCCXHSM module. VCC-DFHSM

```
  does not use it, but if you omit the DD statement for
  ARCSNAP, you will get a "ARCSNAP DD STATEMENT MISSING"
  message in the JES job log when ARCUTIL tries to OPEN it.

Step 2 and Step 3

  You need these steps only if SMF is not going to be used as
  the VCC recording medium.  Step 3 is required if HSM=Y was
  specified for VCC parameters.  The IBM standard utility
  IEBGENER is used to copy the data recorded by VCC to
  Generation Data Groups (GDGs) used by DAILY processing.

  The sample JCL provided and shown above presumes that GDGs
  named '_____.VCC.SMFBKUP' and '____.VCC.HSMBKUP' were
  previously defined.  For information on how to define a
  GDG, see IBM's Access Method Services Reference manual
  (GC26-4019).


Step 4

  This step will execute only if Step 2 and Step 3 run
  successfully.  It is required only if SMF is not used as
  the recording medium for VCC.  This step updates the VCC
  control file indicating that a backup of the data sets
  referenced by DD statements VCCDATA and VCCHSM (if HSM=Y)
  was done.  During initialization processing VCC will check
  the control file to determine if a backup was done.  If no
  backup was done, VCC will not record new information to the
  VCCDATA data set unless BYPASSBKUP=Y is specified for VCC
  parameters.  This ensures that VCC will only record DASD
  space information if the data from the previous VCC run was
  copied successfully.

  Descriptions of the STEPLIB and VCCNTROL data sets are
  under Step 1.
```

### 3.1.2.2 Sample JCL Member VCCRUNH

The following JCL is used for collecting Hierarchical File Systems data.

```
//jobname  JOB  (accounting_data),'VCCRUN',
//         CLASS=A,
//         MSGCLASS=A,
//         NOTIFY=
//*
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*                      - Sample JCL -                          *
//*           CA MICS Space Collector Option (VCC)               *
//*         Hierarchical File Systems Data Collection            *
//*                                                              *
//* Notes:                                                       *
//*   1. VCC requires a minimum of 4096K to run                  *
//*   2. If not included in the System Linklist the STEPLIB      *
//*      must be an authorized library                           *
//*                                                              *
//*                                                              *
//*                                                              *
//* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
//*
//*
//*--------------------------------------------------------------*
//* STEP1  -  Collect HFS data                                   *
//*--------------------------------------------------------------*
//*
//STEP1    EXEC PGM=VCCMAIN,REGION=5000K       <== See Note-1
//*
//STEPLIB  DD   DISP=SHR,DSN=__________        <== See Note-2
//SYSPRINT DD   SYSOUT=*
//VCCSTATS DD   SYSOUT=*
//VCCSNAP  DD   SYSOUT=*
//*
//VCCNTROL DD   DISP=SHR,DSN=______.VCC.DCONTROL
//*
//VCCDATA  DD   DSN=______.VCC.VCCDATA2,
//         DISP=OLD
//*
//*
//*
//VCCPARMS DD   *
HFSSUBT=4
DATALEVEL=(ALL)
/*
```

```
//*
//*-------------------------------------------------------------*
//* STEP2  -  Copy collected data to SMF Daily GDG              *
//*-------------------------------------------------------------*
//*
//STEP2    EXEC PGM=IEBGENER,COND=(4,LT)
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   DSN=______.VCC.VCCDATA2,
//         DISP=OLD
//SYSUT2   DD   DSN=______.VCC.HFSBKUP(+1),
//         UNIT=____,
//         DISP=(,CATLG,DELETE),LABEL=(1,SL)
//SYSIN    DD   DUMMY
//*
//*-------------------------------------------------------------*
//* STEP3  -  Flag VCC Data Copied                              *
//*-------------------------------------------------------------*
//*
//STEP3    EXEC PGM=VCCUTIL,REGION=2048K,
//         COND=(4,LT),PARM='FLAGCOPY'
//SYSPRINT DD   SYSOUT=*
//STEPLIB  DD   DISP=SHR,DSN=__________        <== See Note-2
//*
//VCCSNAP  DD   SYSOUT=*
//*
//VCCNTROL DD   DISP=SHR,DSN=______.VCC.DCONTROL
//*
//
```

VCCRUNH JCL Detail Description

Jobcard

You can run VCC in any job class except a JES3 job class that has JOURNALING specified. You may need to specify a region size if your default size is not at least 4096 K.

Step 1

This step will execute VCC to collect DASD data on Hierarchical File Systems.

STEPLIB - This statement is required only if the VCC modules were not placed in an authorized library that is in the System Linklist. Otherwise, it should point to the authorized library where the VCC modules were moved during installation. (See Chapter 4, VCC Installation).

SYSPRINT - Used for the VCC Message Log. If pre-allocated to a disk data set, DCB attributes should include RECFM=VBA,LRECL=137.

VCCSTATS - Used for VCC Status and Statistics Reports. If pre-allocated to a disk data set, DCB attributes should include RECFM=VBA,LRECL=137.

VCCSNAP - VCC uses this DD statement to dump pertinent storage areas to assist in problem determination.

VCCPARMS - This DD statement is optional. If present, it defines the VCC parameter data set. This data set can be a member of a PDS, a physical sequential file, or a DD *. You can code the parameters as part of the JCL stream.

VCCDATA - This DD statement defines the data set that VCC uses as a recording medium if SMF is not used. This data set must be variable blocked. If you do not specify the BLKSIZE, VCC will set it based on the output DASD device type. For maximum device capacity utilization, you should code the following values on the VCCDATA DD statement.

| Device | Optimum device utilization parameters |
|---|---|
| 3390 | DCB=(BLKSIZE=27998,LRECL=4096,RECFM=VB) |
| 3380 | DCB=(BLKSIZE=23476,LRECL=4096,RECFM=VB) |
| other | DCB=(BLKSIZE=32760,LRECL=4096,RECFM=VB) |

```
  VCCNTROL - Defines the VCC Control File.  Allocated during
  VCC installation, it keeps track of the volumes processed
  and VCC processing timestamps.

  This Control File must be shared across each OS/390 system
  that runs VCC to collect Hierarchical File Systems Data
  within a shared DASD complex.

Step 2

  You need this step only if SMF is not going to be used as
  the VCC recording medium. It uses the IBM standard utility
  IEBGENER to copy the data recorded by VCC to a Generation
  Data Group (GDG) used by CA MICS Daily processing.

  The sample JCL provided and shown above presumes that a GDG
  named '_____.VCC.HFSBKUP' was previously defined.  For
  guidance on how to define a GDG, see IBM's Access Method
  Services Reference manual (GC26-4019).

Step 3

  This step will execute only if Step 2 ran successfully.  It
  is required only if SMF is not used as the recording medium
  for VCC.  This step updates the VCC Control File indicating
  that a backup of the data set referenced by the VCCDATA DD
  statement was done.  During initialization processing VCC
  will check the Control File to determine if a backup was
  done.  If no backup was done, VCC will not record new
  information to the VCCDATA data set unless BYPASSBKUP=Y is
  specified for VCC parameters, or a new data set is used.
  This ensures that VCC will only record DASD space
  information if the data from the previous VCC run was
  copied successfully.

  Descriptions of the STEPLIB and VCCNTROL data sets are
  under Step 1.
```

### 3.1.2.3 Setting Up Hourly Scans with VCC

This section describes how to process data created by DATAINFO=N in VCC (also see section 3.1.4 for more discussion on running VCC at frequent intervals).

You may need to examine public volumes, work storage groups or pools at more frequent time intervals than once per day. While a DAILY inventory of volumes containing "normal" data sets (those with a permanent DSNAME) is sufficient and efficient for accounting purposes, you may need an hourly inventory of public volumes to see the change in volume free-space.

The VCAARCT routines will not be able to determine "ownership" of the public volumes because of the format. You can use the DATAINFO run-time parameter to do this. Setting DATAINFO=N will filter out data set information and provide the basis for the type of volume observations needed to track capacity of special volumes like this.

The following contains some requirements and suggested ideas. We try to convey as much as possible by example for conciseness. Translate these general ideas into your own requirements. DATAINFO=N is best used with:

- o INCLUDEVOL=(WORK*,HSM*) (Scratch pools and/or HSM ML1 vols)
- o VSAMSUBT=0
- o a separate VCCNTROL file
- o a separate VCCDATA file
  - allocate a small amount of tracks since only one record per volume is recorded
  - allocate VCCDATA to a temporary data set on SYSDA
  - follow VCCNTRL execution step with IEBGENER which copies VCCDATA to an output data set with DISP=MOD.

```
Run frequency should be every 30 minutes or 1 hour.

This process reads VTOCs and filters out all data set level
information and writes only one VCC record per volume
scanned. Use the following techniques to obtain this
information:

____ 1. The hourly inventory must use SMFID=xxx to
        differentiate the output from the daily inventory.
        You can use SMFID=194 for the hourly, while the daily
        defaults to SMFID=198.

        The hourly inventory should write to a temporary data
        set then MOD onto a permanent file for HOURLY data.

        Example:

        //S1 EXEC PGM=VCCNTRL
        //VCCDATA DD DISP=(,PASS),DSN=&&HOUR,
        // SPACE=(TRK,(5,1),RLSE)
        ..
        .. .... remainder of VCC JCL .....
        ..
        //S2 EXEC PGM=IEBGENER
        //SYSUT1 DD DISP=(OLD,DELETE),DSN=&&HOUR
        //SYSUT2 DD DISP=MOD,DSN=VCC.VCCDATA.HOURLY

        When the daily inventory runs, it should do some
        cleanup on behalf of the hourly inventory.

        - VCC.VCCDATA.HOURLY (in the MOD file above) should
          be copied into another data set to isolate it from
          subsequent hourly scans.

          Example:

        //S1 EXEC PGM=VCCNTRL   **for daily inventory**
        ..
        .. .... remainder of VCC JCL .....
        ..
        //S2 EXEC PGM=IEBGENER  **cleanup for hourly data**
        //SYSUT1 DD DISP=(OLD,DELETE),DSN=VCC.VCCDATA.HOURLY
        //SYSUT2 DD DISP=SHR,DSN=VCC.VCCDATA.HOURS
        //S3 EXEC PGM=IEFBR14  **setup a new place to MOD**
        //NEWPLACE DD DISP=(,CATLG),DSN=VCC.VCCDATA.HOURLY,
        // SPACE=(TRK,(5,5),RLSE)
```

____ 2. Modify SP.GENLIB VCAGENIN to add a special SMF
record number for the hourly scan using the INPUTSMF
statement.

Run VCAGEN.

____ 3. Concatenate the hourly inventories for the whole day
with a single daily inventory for the same day and
run them both into the same DAY90 step (in the same
unit database). Example:

```
//INPUTSMF DD DISP=SHR,DSN=VCC.VCCDATA.HOURS
//         DD DISP=SHR,DSN=VCC.VCCDATA.DAILY
```

____ 4. Code the _USRSEL exit as follows to identify the
hourly data and change it in a way that allows for
later separation of the data at reporting time.

For example:

```
IF ROUTINE EQ 'DYVCAFMT' AND
  SMFRTYPE EQ 194 THEN DO;
  ORGSYSID='HOUR'; /*variable tested at report time*/
  SMFRTYPE=198;    /*treat the rest of the data*/
                   /*the same*/
END;
```

____ 5. Running hourly scans by performing the above steps
has the following effect on the VCAVOA file:

All volumes will have one observation with the same
ENDTS (as each other) as a result of the DAILY scan.

Volumes in the hourly scan domain (via INCLUDEVOL)
will have 24 additional observations in the same
cycle at DETAIL timespan.

One of those hourly observations will occur in the
same hour of the day as the sole observation from the
DAILY scan.

And it has the following effect on reporting (example is PROC CHART):

```
DATA EXTRACT;        /* Create hourly-obs only file*/
SET DETAIL.VCAVOA01;
 IF ORGSYSID='HOUR';
 PROC PLOT DATA=EXTRACT;
  PLOT VOASPACA*HOUR;  /*Now hour has more than one*/
                       /*obs*/
   BY VOLSER;
```

StorageMate has a built-in report to handle this (see STGEJE).

## 3.1.3 Job Operational Flow

This section describes the operation logic flow of the VCC job with respect to the integrity of the data that VCC generates.

When recording to SMF:

When VCC completes processing for a volume, a volume total record is generated and written. If VTOC processing does not complete normally, this record is not written. During VCA processing, any data set and catalog records will be ignored if they represent a volume for which no total record was found. This makes sure that data for partially processed volumes is ignored by VCA.

When recording to a data set:

The VCC job collects DASD space utilization data. The control file is used to time stamp the VCC processing start time, keep track of previous process start time and volumes processed, and to make sure that backups are done if SMF is not chosen as the VCC recording medium.

The VCCDATA data set is a reusable file in which VCC will record its SMF records. VCC ensures that the disposition for this data set is always OLD. MOD is not allowed.

The control file is not updated until the end of the VCC data collection process (this step). Therefore, if this step does not complete successfully it can be rerun any number of times until a successful step termination occurs. Because the VCCDATA data set is not used as input to the CA MICS Data Base, CA MICS cannot process a partial data collection run. And because the VCCDATA data set is reusable, CA MICS will see only the latest data and not data from several VCC runs strung together.

When the control file is updated, a flag is set that indicates that a backup (STEP2) is needed. If a backup is not successfully done, then subsequent VCC runs will fail if this flag is still on. The last step (STEP3) resets this flag if the condition code for STEP2 is 0.

### 3.1.4 Scheduling Considerations with CA MICS and DFHSM

VCC is designed to be run once a day to take a snapshot of the DASD configuration. This snapshot, in the form of SMF records, is then fed into the CA MICS Space Analyzer Option and Accounting and Chargeback products during the CA MICS daily run.

VCC can be run at any time of the day, but it must be run before the CA MICS daily run. It should not be run during daily backup and restore processing because this type of processing slows the VCC run significantly.

To ensure that VCC is run at least once a day, we advise that you do one of the following:

- Have VCC started automatically through the JES2 automatic command facility at the selected time.
- Have operations run VCC as part of their daily schedule or use a production control package.

Either approach suggests it may be worthwhile to set up VCC as a started task (STC) by creating a PROC in a PROCLIB data set so the JCL can be invoked with an operator START command. Use of symbolic parameters and a scheduling package will enable you to present VCC with a different set of run time parameters in the VCCPARMS data set according to the variables such as:

- day of week (for example, CATCHECK=Y only on Sun.)
- primary MVS image vs. secondary images (shared DASD) (primary MVS scans all online DASD while secondary MVS images scan with a concatenated VCCPARMS data set containing EXCLUDEVOL lists)

VCC must not be run in a JES3 jobclass that has JOURNALING specified. A system 130 abend will result because of a conflict in device ownership.

If you have DFHSM implemented and add HSM=Y to collect information on migrated data sets and/or backup versions, you should ensure that VCC runs and completes either just BEFORE or just AFTER DFHSM Daily Space Management is run. Because VCC is so fast, it is probably better to run VCC before HSM daily processing. That way, VCC has finished data collection; then CA MICS DAILY processing and DFHSM daily processing can proceed in parallel to each other. The important point regarding the VCC-DFHSM interface is consistency. The statistics developed in the DFHSM files in the CA MICS database will be more useful if the inventory view of DFHSM-managed storage is always taken at the same time relative to the data management actions performed by the DFHSM daily (for example, MIGRATE, DELETE, BACKUP, etc.).

Information on two examples of uses of the VCCPARMS keywords are in the following subsections:

1 - Hourly Collection of Volume Data (DATAINFO)
2 - Weekly Collection of Catalog Status (CATCHECK)

### 3.1.4.1 Hourly Collection of Volume Data (DATAINFO)

This option allows you to take an inventory of the volumes on a more frequent basis than the entire inventory you take at the data set level.

Assuming you would find it useful to collect data about the changes in key volume level statistics on an hourly basis during prime shift hours you could "clone" your production run setup and make several modifications to support this special type of data collection:

- o Establish VCC as a separate PROC in a PROCLIB so a START command can be automatically issued hourly by JES2 or by a production scheduling package.

  The JCL in this PROC must be different from the JCL in the PROC for the "daily" version of VCC. Key differences are the following:

  - The "hourly" PROC must ALWAYS allocate a different VCCNTROL file to preserve the DURATION calculations needed for data set level processing.

  - The "hourly" PROC should allocate VCCDATA to a GDS (generation data set).

  - The "hourly" PROC should allocate a VCCPARMS data set that specifies:

    ```
    DATAINFO=N   (VCC writes only 1 record per VOLSER)
    HSM=N        (only collect DFHSM inventory daily)
    SERIALIZE=N  (no RESERVEs on VTOCs)
    CATCHECK=N   (no LOCATEs to slow down scanning)
    VSAMSUBT=0   (no reading catalogs or VVDSs)
    VTOCSUBT=16  (specify a high level of multitasking)
    ```

  INCLUDEVOL or EXCLUDEVOL statements are allowed to subset the data if desired. For example, all volumes belonging to a storage group in which only temporary data sets are allocated could represent a pool of volumes you would like to monitor hourly with this method. INCLUDEVOL is a facility for subsetting the scanning accordingly along with the above list.

### 3.1.4.2 Weekly Collection of Catalog Status (CATCHECK)

This approach illustrates another variation in the set of VCCPARMS keywords that fulfills special operational needs and assists the storage administrator in identifying problem data sets.

Let's assume you would find it useful to choose one VCC run per week to examine the catalog status of VTOC data sets. The CATCHECK=Y option will perform this check but it is not worth the extra processing incurred to perform this extra function every day. Therefore, every week or two, perhaps on a weekend night, it is useful to add the CATCHECK function to your production run.

A "weekly" VCC PROC for catalog check processing should take the following into consideration:

- The "production" VCCDATA and VCCNTROL data sets can be used, because this is essentially the same data collection process as the "production" run.

- If HSM=Y is requested in "production", it should be retained in this "weekly" run too.

- Serialization of the VTOCs is not allowed during CATCHECK processing. You may want to reduce the level of VSAM and VTOC subtasking in the "weekly" run too. Expect longer run times for the job but the same magnitude of records written to VCCDATA.

- Be aware that the LOCATE done by VCC with the CATCHECK option will be subject to the variety of ways enterprises can connect user catalogs and share DASD in today's complex environments.

  For example, you may want to consider the use of carefully selected EXCLUDEVOL lists when running with CATCHECK=Y. For example, if the IPL volume for SYSB is online and shared with SYSA, all SYS1. data sets cataloged in the master catalog on SYSB will be accessible by VCC but naturally, LOCATE processing will indicate that these SYS1. data sets are cataloged to a different volume.

Therefore, if each weekend you presented your production VCC run with a VCCPARMS setup that requested CATCHECK=Y and EXCLUDEVOL=(RESBRS), you should know that the DURATION for all data sets on that EXCLUDED volume will be properly adjusted when the next VCC run in production performs the scan of RESBRS. This is because the control file (VCCNTROL) is not updated if a volume is EXCLUDED. In this way, if the normal span of time between scans is approximately 24 hours, this will be adjusted on a volume basis to (approximately) 48 hours if a volume is EXCLUDED for a day.

## 3.1.5 VCC Reporting

The section describes the types of reports produced by the CA MICS Space Collector Option.

For the standard DASD collection, the following ddnames will contain reports:

- VCCVTMSG and VCCVSMSG - VCC Scan Logs
- VCCSTATS - VCC Run Status/Run Statistics
- VCCHSMSG - The DFHSM Interface Report

For the HFS collection the following ddnames will contain reports:

- SYSPRINT - USS File System Message Log
- VCCSTATS - Uss File System Run Statistics Reports

### 3.1.5.1 VCC Scan Logs

The VCC Scan Log is generated by VCC as the VTOC and catalog scans are being done. The purpose of this report is to log unusual situations that may occur during the scans. The message descriptions in Chapter 7 for the VCCVTOC, VCCVCNTL, and VCCVSCAN modules list the informational messages that can be generated.

Figure 3-2 shows a partial typical standard VTOC Scan Log.

Figure 3-3 shows a partial typical standard Catalog Scan Log. From this log, you can determine the catalogs that are being excluded from processing because they are on mountable volumes.

### 3.1.5.2 VCC Run Status/Run Statistics

The following reports generated by the standard DASD collection process are described below:

- VCC Initialization and Parameter Message Log
- VCC Processing Summary
- VTOC Scan Summary
- Catalog Scan Summary
- IBM DFSMS Storage Group Summary

The following reports generated by the USS File System Collector are described below:

- Processing Summary
- File System Summary

VCC Initialization and Parameter Message Log

This section of the VCC Run Status Report is produced during standard VCC initialization. VCCNTRL initialization error messages, an echo of the run-time parameters, and parameter scan error messages appear in this section. See the VCCNTRL and VCCPARSE message descriptions in Chapter 7, Return Codes and Messages, for a list of these messages.

Figure 3-1 shows a typical Initialization and Parameter Message Log.

VCC Processing Summary

This section of the VCC Run Status Report is produced by both the standard and USS File System collection process just prior to termination.

Figure 3-4 contains a typical example of the standard VCC Processing Summary. Figure 3-10 contains a typical example of the Processing Summary for the HFS collection process. The following describes the entries produced by both collection processes.

Catalog Scan Elapsed Time - The total elapsed time, in hh:mm:ss.th format, it took VCC to process the master and all user catalogs on the systems that were selected to be scanned.

Central Processors - The number of central processors online

```
                and available for use on the z/OS image that
                VCC was executing.

Collection CPU Time - The total amount of CPU usage time.

Collection Duration - The total elapsed time in hours,
                minutes, seconds, and hundredths of seconds
                that VCC took to perform the SCAN process.

Collection Error Messages Issued - The number of error
                (TYPE=E) messages contained in the VCC
                Message log.

Collection Highest Completion Code - The highest condition
                code passed to z/OS by VCC.

Collection Warning Messages Issued - The number of warning
                (TYPE=W) messages contained in the VCC
                message log.

Collector Records Written - The number of records that VCC
                wrote.  These records are either written to
                SMF or to a data set (user-selectable VCC
                option).

Collector Run Number - The internal number generated by VCC
                for identifying each unique collection
                execution.

Collector Subtasks Started - The total number of subtasks
                started by VCC to perform the scan process.

Collector Subtask Buffer Wait Count - The total number of
                times that a VCC subtask had to wait for
                output buffers. Use this value to determine
                the value of the NUMWTRBUF VCC run-time
                parameter.

DASD Devices Online - The total number of DASD devices in the
                I/O configuration of the system on which VCC
                executed.  This is the number of devices in
                the volume table built by VCC initialization
                (see Chapter 5, Operation).

DASD Devices Scanned - The count of the number of DASD
                devices in the I/O configuration that were
                eligible to be processed by VCC.  This takes
                into account INCLUDE/EXCLUDEVOL.  It is the
                number of DASD devices in the VSCT (see
```

```
                Chapter 5, Operation).

Free Extent Thresholds - Displays the distribution threshold
                values specified by the user or defaulted
                by the system.  These values are used to
                provide a distribution analysis of the
                amount of free space on a DASD volume.  Each
                value represents the maximum number of DASD
                tracks for inclusion in the threshold level.

HFS  Data Sets Detected - The number of Hierarchical File
                System data sets that VCC detected during
                the VTOC scan process.

ICF BCS  Catalog Records - The number of ICF BCS records VCC
                read during the catalog scan process.

ICF BCS  Catalogs Scanned - The number of ICF Basic Catalog
                Structure (BCS) data sets that were scanned
                by VCC.

ICF BCS  NonVSAM Records - The number of ICF BCS records
                related to non-VSAM catalog entries VCC read
                during the catalog scan process.

ICF VVDS Catalog Records - The number of ICF VVDS records VCC
                read during the catalog scan process.

ICF VVDS Catalogs Scanned - The number of ICF VSAM volume
                data sets (VVDSs) that were scanned by VCC.

ICF VVDS NonVSAM Records - The number of ICF VVDS records
                related to non-VSAM entries VCC read during
                the catalog scan process.

IPL Volume     - The volume serial number of the system
                resident DASD volume.

Max DSCB Extent  Queue Length - The largest number of VTOC
                Format 3 DSCB entries in the DSCB resolution
                queues at any one time.

Max DSCB Dataset Queue Length - The largest number of VTOC
                DSCB chain resolution entries in the DSCB
                resolution queues at any one time.

Operating System Level - The version level and function
                modification ID (FMID) of the operating
                system on which VCC was executed, displayed
```

primarily to provide information to Technical Support, if needed. For assistance, contact Technical Support at http://ca.com/support.

PDSE Data Sets Analyzed - The number of PDS/E type data sets that were successfully processed during the VTOC scan process.

PDSE Data Sets Detected - The number of PDS/E type data sets that VCC detected during the VTOC scan process.

Physical Processors Detected - The number of central processors online and available for use on the z/OS image that VCC was executing.

Previous Collection Run On System - The name of the last z/OS system that ran VCC.

Previous Collection Date - The date and time of the last successful VCC run. The timestamp shown is placed in the VCC record and is used by the CA MICS Space Collector Option to compute the duration to use for DASD space charging in Accounting and Chargeback. If the VCC control file is new or null at the time of the VCC run, the word "UNKNOWN" is placed on this line and the CA MICS Space Collector Option computes a default duration based on VCA parameter generation values.

Previous Process Time - The VCC processing start time and Julian date for the previous VCC run.

Note: If the last VCC run ended with condition code zero although no records were produced (which may or may not indicate an error), this timestamp is updated. The timestamp shown is placed in the VCC record and is used by the CA MICS Space Collector Option to compute the duration to use for DASD space charging in Accounting and Chargeback. If the VCC control file is new or null at the time of the VCC run, the word "UNKNOWN" is placed on this line and the CA MICS Space Collector Option computes a default duration based on VCA parameter generation values.

```
Processing Start Time - The VCC processing start time and
                        Julian date.

SCP FMID              - The function modification ID (FMID) for z/OS
                        is shown for technical support purposes.

SMFID                 - The SMF ID of the z/OS image on which VCC
                        executed.

SMS Level             - The version, release, and modification level
                        of the Data Facility Product (DFP) or System
                        Managed Storage (SMS) for technical support
                        purposes. The SMS level is shown in the
                        format vv.rr.mm.

SMS or DFP Level      - The version, release, and modification
                        level of the Data Facility Product (DFP) or
                        System Managed Storage (SMS) for technical
                        support purposes. The SMS level is shown in
                        the format vv.rr.mm whereas the DFP level is
                        shown in the format v.r.m.l.

SYSID                 - The SMF ID of the z/OS image on which VCC
                        executed.

Sysplex Name          - The eight-character name of the z/OS sysplex
                        of which this system image is a member.

System Name           - The eight-character name of the z/OS image
                        on which VCC executed.

System Control Program - The level of z/OS on which VCC was
                        executed, displayed primarily to provide
                        information to Technical Support, if
                        needed.  For assistance, contact Technical
                        Support at http://ca.com/support.

Temporary Datasets Auto-Excluded - IF TEMPDSN=N is in effect
                        for the VCC run, the count shown indicates
                        the number of records VCC did not write to
                        its output data set (VCCDATA).

Total Online DASD Devices - The total number of DASD devices
                        in the I/O configuration of the system on
                        which VCC executed.  This is the number of
                        devices in the volume table built by VCC
                        initialization (see Chapter 5, Operation).
```

```
Uncataloged NonVSAM Datasets - If CATCHECK=Y is in effect for
                the VCC run, the count of data sets that
                were found to be either uncataloged or
                invalidly cataloged is shown here.  When
                CATCHECK=N is in effect, a question mark (?)
                is shown to indicate that this statistic is
                not known from this VCC scan.

USS File Systems Detected - The number of accessible file
                systems detected by VCC.

USS File Systems Monitored - The number of accessible file
                systems that VCC generated data for.

USS File System Files Detected - The number of files within a
                file system that VCC found.

USS File System Directories - The number of directories
                within a file system that VCC found.

VCC Records Bypassed - The number of records that were not
                written due to a validity check on the
                record length done by VCCWRITR.  In some
                cases, a bad catalog cell causes a record of
                excessive length to be constructed and
                passed to VCCWRITR.  Rather than abend,
                VCCWRITR will SNAP a portion of the record
                (up to the maximum LRECL in the DCB) and
                count the record as bypassed. Normally, the
                count that appears here should be zero. If
                non-zero, you can usually examine the SNAP
                output and locate the data set name.  Other
                facilities such as IDCAMS DIAGNOSE or PDF
                can be used to determine if a data set or
                catalog entry is invalid.

VCC Records Written - The number of records that VCC wrote.
                These records are either written to SMF or
                to a data set (user-selectable VCC option).

VSAM Catalog Records - The number of VSAM catalog records VCC
                read during the catalog scan process.  This
                number includes direct reads that were done
                to get information from catalog extension
                records.  Therefore, a catalog record can be
                counted twice.

VSAM Catalogs Scanned - The number of standard VSAM catalogs
                that were actually processed by VCC.
```

VTOC Dataset Records - The number of VTOC DSCBs that VCC read during the VTOC scan process.

VTOC Scan Elapsed Time - The total elapsed time, in hh:mm:ss.th format, it took VCC to process the VTOCs on the DASD devices that were selected to be scanned.

zSeries File Systems Detected - The number of zFS type data sets that VCC detected during the VTOC scan process.

VTOC Scan Summary

This section of the VCC Run Status Report is a list by DASD device unit address of every DASD volume in the I/O configuration whose device type VCC supports. This report contains the following information:

VOLUME - The volume serial number of the DASD device.

UNIT ADDR - The unit address of the DASD device.

DEVICE TYPE - The device type of the DASD device.

Figure 3-7 shows an example of the VTOC Scan Summary.

The following information will be present only if the VTOC of the DASD device was scanned by VCC.

SCAN ELAPSED TIME - The elapsed time in seconds it took VCC to scan the VTOC on the device.

%TOTAL SPACE USED - The amount of storage space allocated on the volume expressed as a percentage.

TOTAL VTOC TRKS - The size of the VTOC in tracks.

DSCB USE STATISTICS - A set of DSCB counts that are useful in determining:

- VTOC over-/underallocation
- VTOC DSCB spread (how many DSCBs are used in relation to the high used DSCB)

TOTAL - The total number of preformatted 140-byte

```
                DSCBs in the VTOC.

USED          - The number of DSCBs in the VTOC that are in
                use (not binary zeros).

F1            - The number of Format 1 (data set identifier)
                DSCBs in the VTOC.  The sum of this number
                and the number of Format 8 DSCBs below
                represent the number of data sets on the
                DASD device.

F3            - The number of Format 3 (Format 1 data set
                extents greater than 3) DSCBs in the VTOC.

F5            - The number of Format 5 (volume free space
                extents) DSCBs in the VTOC.

F7            - The number of Format 7 (extended free space
                extents) DSCBs in the VTOC.

F8            - The number of Format 8 (data set identifier)
                DSCBs in the VTOC.  The sum of this number
                and the number of Format 1 DSCBs above
                represent the number of data sets on the
                DASD device.  The Format 8 DSCB is used for
                data sets that are eligible for placement in
                cylinder-managed space on an Extended
                Address Volume (EAV).

F9            - The number of Format 9 (data set attribute
                information) DSCBs in the VTOC.  These DSCBs
                occur only in conjunction with Format 8
                DSCBs for data sets eligible for placement
                in cylinder-managed space on an EAV.

HIGH USED     - The DSCB number in the VTOC of the last DSCB
                in use.  This number is equivalent to the
                number of VSAM READ macros issued by VCCVTOC
                to read the VTOC.


The fields in the next set indicate specific attributes of
the DASD volume.  "Y" indicates that the attribute is
available or true.  "N" indicates that it is unavailable.
"?"  indicates unknown or unable to determine if the specific
attribute is available.

V I X   - Specifies whether a VTOC index exists on the DASD
          volume.
```

V V D - Specifies whether an ICF VVDS exists on the DASD volume.

I C F - Specifies whether an ICF BCS exists on the DASD volume.

S M S - Indicates the DFSMS ownership status of this volume. If an "I" appears in this field, the volume is in the process of being converted to SMS. The volume status is "INITIAL," indicating that some data sets are under SMS control.

S H R - Indicates whether the DASD device is identified to the operating system as a device that is sharable between multiple systems.

E A V - Indicates whether the DASD device is identified to the operating system as an Extended Address Volume (EAV). An EAV volume is one that is configured with more than 65,520 cylinders. Cylinder-managed space exists and extended attribute DSCBs, format 8 and 9 DSCBs, are permitted in the VTOC.

STORAGE GROUP - If SMS is active and the volume is SMS-managed, the storage group name is shown.

PROCESS STATUS - Indicates error situations (if any) that occurred during the scan of a volume. This report field will be present for all DASD devices in this section of the VCC Run Status Report.

The possible status values are:

Device Not Ready - VCC bypassed the processing of this device because the device was not ready at the time the VTOC scan control table (VSCT) was built.

DSN Residuals - One or more queue elements for Format 1 DSCBs were found during end of VTOC processing. This condition can only occur when an error condition is encountered that causes premature termination of scan processing for a DASD volume.

EXT Not Found - A locate request for a Format 3 DSCB on the EXT queue resulted in a not-found condition. This may be caused by an invalid pointer in a Format 1 or 3 DSCB as the result of an interrupted VTOC update process. The scan result may or may not be in error depending on the condition of the VTOC.

EXT Residuals - One or more residual Format 3 DSCB extent records were read, but the associated Format 1 DSCB records for them had not been read by the VCC scan task. This condition will most likely occur when processing DASD volumes with extremely high activity if the SERIALIZE=N option is in effect for the VCC scanner. This condition might also occur as a result of a system crash that occurred during an update of the VTOC that has resulted in a corrupted VTOC.

Incomplete Scan - A DASD volume being processed encountered an error condition. One or more data records have been written to the volume record free extent segment, but not all of the VTOC has been successfully processed.

Lost Tracks - VCC VTOC processing of this volume could not account for all of the tracks on the volume. The sum of the allocated tracks, free tracks, size of VTOC, and 1 for track 0 did not equal the track capacity of the volume.

Migration Level 1 - The DASD volume being scanned was determined to be a DFHSM migration level 1 storage volume. Only the VTOC Volume Record will be output for this volume, regardless of the value of the DATAINFO= parameter.

```
Mount Attributes - VCC processing of this
     device was bypassed because the
     device was not mounted in a manner
     that satisfied the requirements of
     VCC (online and resident).

No VTOC Found - VCC was unable to process
     a DASD volume because a valid pointer
     to the VTOC did not exist in the UCB
     for the device.

Processing Error - An error occurred
     during the processing of the VTOC on
     the DASD device.  The VTOC scan log
     will contain a description of the
     error.

PDSE Extended - The DASD volume was
     explicitly specified on an
     EXTENDEDPDSE parameter statement
     causing extended attribute data for
     the PDS/Es on the volume to be
     collected.

PDSE No Extended - The DASD volume was
     explicitly specified on an
     NOEXTENDEDPDSE parameter statement
     causing extended attribute data for
     the PDS/Es on the volume NOT to be
     collected.

Record Overflow - A volume contained more
     free space extents than could be
     placed into the data record. The VTOC
     Volume Record free extent segment
     will be incomplete.

Reserve Bypassed - A hardware RESERVE was
     not done for this volume before it
     was processed.  This was because
     either the volume was specified in a
     BYPASSRES parameter or because the
     RESRETRY limit was exceeded and
     RESFAILOK=Y was coded or defaulted.

Scan Abend Sxxx - An abend occurred during
     scan processing of a DASD volume.
     The Sxxx value indicates the abend
     code returned by the system.  Scan
```

processing for the volume is terminated and one of two actions is taken. If the abend code is determined to be recoverable, such as an S213-04 from an invalid VTOC, processing for the volume is terminated, the environment is cleaned up, and processing continues with the next DASD volume. If the abend is determined to be non-recoverable, processing for the volume is terminated and the scan subtask terminates after cleanup processing.

Status Bypassed - The UCB status of the DASD volume was found to contain indicators that would prevent successful scan processing of the volume. If this condition persists for three successive attempts, the volume is permanently bypassed for this processing execution.

Volume Excluded - VCC bypassed processing of this device because the device either was not INCLUDEd or was specifically EXCLUDEd.

Volume Not Read - A VTOC was not processed because of a critical error.

Volume Reserved - VCC could not process this volume because it was reserved by another user and VCC could not reserve it. You will receive this message if the RESRETRY limit was exceeded and RESFAILOK=N was coded.

VM Volume - Indicates a DASD volume that has been formatted for use by a VM operating system. These volumes either have no VTOC or a non-standard VTOC that cannot be processed by the VCC scan task.

Figure 3-7 shows an example of a VCC Run Status Report.

```
Catalog Scan Summary

This section of the VCC Run Status Report lists the catalogs
that were processed by VCC.  The following information is
given:

CATALOG NAME    - The name of the catalog that was scanned.

VOLUME          - The volume serial number of the DASD device
                  on which the catalog resides.

UNIT ADDR       - The unit address of the DASD device on which
                  the catalog resides.

CATALOG TYPE    - The type of catalog.  This value can be:

                    ICF  - for ICF BCS Catalogs
                    VVDS - for ICF VVDS Catalogs
                    VSAM - for VSAM catalogs

CATALOG RECORDS READ - The number of records that were read
                  from the catalog by VCC.

CATALOG ENTRY TYPES - The number of entries in the catalog
                  that were actually processed by VCC.  It can
                  be any of the following:

                    VSAM CLUSTER - the number of VSAM clusters
                                   in the catalog.

                    AIX          - the number of alternate
                                   indexes in the catalog.

                    CATALOG      - the number of user catalog
                    CONNECTOR      connector entries in the
                                   catalog that VCC actually
                                   processed.

                    NONVSAM      - the number of non-VSAM
                                   entries in the catalog.

CATALOG PROCESS TIME - The total elapsed time in seconds it
                  took VCC to scan this catalog or VVDS.

PROCESS STATUS - The error situations that occurred during
                  the scan of the catalog.  The possible
                  values are:

                    * Not Read * - A BCS, VVDS or VSAM catalog
```

was not processed because of a critical error.

Allocation Error - The catalog could not be allocated. Check the Catalog Scan Log for the allocation error return and reason codes. Refer to the applicable IBM publication for information regarding the return and reason codes.

D/I S-O-F Error - An error occurred during the processing of a VSAM catalog record. The data/index cell sets-of-fields information is not accurate.

EXT S-O-F Error - An error occurred during the processing of a VSAM catalog record. An extension record's sets-of-fields information is not accurate.

Locate Error - The catalog could not be located via SVC 26 (LOCATE).

MODCB Error - An error occurred during the execution of a VSAM MODCB macro. Check the Catalog Scan Log for the MODCB error return and reason codes. Refer to the applicable IBM publication for information regarding the return and reason codes.

Open Error - An error occurred during the OPEN of a VSAM catalog. Check the VCC JOBLOG for VSAM error messages. Refer to the applicable IBM publication for information regarding the return and reason codes.

Point Error - An error occurred during the execution of a VSAM POINT macro. Check the Catalog Scan Log for the error return and reason codes. Refer to the applicable IBM publication for information regarding the return and reason codes.

Processing Error - Scan processing for a BCS, VVDS. or VSAM catalog did not complete. The interruption was probably due to an abend condition. Refer to the Catalog Scan Log for additional information.

Read Error - An error occurred during the execution of a VSAM GET macro. Check the Catalog Scan Log for the error return and reason codes. Refer to the applicable IBM publication for information regarding the return and reason codes.

SMF Write Error - An error condition code was returned from the SMFWTM service while attempting to write the VCC Catalog record to SMF. Refer to the Catalog Scan Log for additional information.

Unknown Status - An error condition was encountered while processing a BCS, VVDS, or VSAM catalog. The status information returned was unknown to the scan task. Refer to the Catalog Scan Log for additional information.

Figure 3-5 shows an example of the Catalog Scan Summary.

IBM DFSMS Storage Group Summary

This section of the VCC Run Status Report contains a list of all the DFSMS storage groups found on the system that ran VCC. This report contains the following information:

STORAGE GROUP NAME - The name of the DFSMS storage group.

STATUS - The status of the storage group at the time of the VCC volume VTOC scan (that is, ENABLED, DISABLED, QUIESCED, and so on).

TOTAL SPACE (TRACKS) - The total storage capacity in tracks for the storage group.

PERCENT USED - The percentage of space used for the

storage group.

VOLUME COUNT - The total number of DASD volumes found that belong to the storage group.

VOLUME LIST - A list of DASD volumes that belong to the storage group.

Figure 3-6 shows an example of the DFSMS Storage Group Summary.

### 3.1.5.3 The DFHSM Interface Report (DDname VCCHSMSG)

The VCC DFHSM Interface for the standard DASD collection produces a report similar to the one shown in Figure 3-8. All processing messages and statistics related to HSM are isolated on this file (DD statement is VCCHSMSG) rather than being added to VCCSTATS.

This SYSOUT type file provides an independent place to log out any error messages related to the VCC-DFHSM processing and prints record counts by record subtype to show the amount of data written to VCCHSM or to SMF. Also shown on this report are processing statistics such as the elapsed time of the current run and the time and date of the previous run.

### 3.1.5.4 VCC HFS Message Log

The VCC HFS Message Log is generated by VCC during the HFS collection process. The purpose of this report is to log messages and information pertaining to unusual situations that may occur during the collection process. During VCC initialization processing, the runtime parameters specified are listed in this log.

For a description of the messages shown in this report, see Chapter 7.

Figure 3-9 shows a typical HFS Message Log.

### 3.1.5.5 USS File System Run Statistics Report

The USS File System Run Statistics Report produces a Processing Summary and a File Systems Summary. The Processing Summary has already been described in Section 3.1.5.2. Figure 3-10 contains a typical example of the Processing Summary for the USS File System Collector process.

File Systems Summary

This section of the VCC Run Statistics Report contains a list of all the accessible file systems that VCC detected during its processing. This report contains the following information:

FILE SYSTEM NAME - The data set name of the USS file system.

T - Identifies the USS file system type that is being reported. The following USS file system types are possible:

H - Hierarchical File System (HFS)
T - Temporary File System (TFS)
Z - zSeries File System (zFS)

VOLUME - The DASD Volume serial that contains the file system. If the volume serial starts with 'V@' the DASD is a virtual volume.

DEVICE TYPE - The DASD Device type of the volume where the file system resides.

UNIT ADDR - The unit address of the DASD device.
XPT: - File System is exported to another platform.

MOUNT POINT - The initial directory name that is the starting point of the total file structure available on this z/OS system.

Figure 3-11 contains an example of a typical File Systems Summary.

| CA MICS | Space Collector | Version 3.2.1 @ VCC6690 | Page: 1 |
|---|---|---|---|

```
Run Status Report: DDNAME(VCCSTATS)                                          Copyright (C) 2013 CA. All Rights Reserved.

Initialization and Parameter Message Log
----------------------------------------
  DataInfo=Y
  FXThreshold=(15,150,750,1500,7500,15000,75000)
  HSM=Y
  HSMBackup=N
  HSMMigrate=Y

  Serialize=N

  SReport=Y
  VsamErr=3
  VsamSubt=10
  VtocErr=3
  VtocSubt=20

18:24:55.5 VCCCN03I DASD Space Collector initialization initiated

18:24:55.5 VCCCN40I ControlFile data set on: Volume(TSU033) Dataset(MDI.VCC6680.VCCAFD.VCNTL)
18:24:55.5 VCCCN41I ControlFile attributes: Dsorg(PS) Recfm(VB) Lrecl(6160) Blksize(6164) Bufno(5)
18:24:55.5 VCCCN42I Collection data set on: Volume(DMP057) Dataset(MDI.VCC6680.VCCAFD.VDATA)
18:24:55.5 VCCCN43I Collection attributes: Dsorg(PS) Recfm(VB) Lrecl(27994) Blksize(27998) Bufno(30)

18:24:55.5 VCCCN04I    12,524 DASD devices are defined
18:24:55.5 VCCCN04I    10,395 PAV  base devices are defined
18:24:55.5 VCCCN04I     2,128 PAV  alias devices are defined
18:24:55.6 VCCCN02I     5,507 DASD devices are online
18:24:55.6 VCCCN05I    3,296K Obtained for Volume Table
18:24:55.6 VCCCN05I    1,140K Retained for Volume Table

18:24:55.7 VCCCN48I Load module level information
18:24:55.7 VCCCN49I BLDVT: VCC6680 CATQM: VCC6680 CVAF:  VCC6680 ETXR:  VCC6680 FAMS:  VCC6680 NTRL:  VCC6680 PARSE: VCC6680
18:24:55.7 VCCCN49I QSCAN: VCC6680 SYMBM: HBB7707 UEXIT: N/A     VBCS:  VCC6680 VCNTL: VCC6680 VSCAN: VCC6680 VTOC:  VCC6680
18:24:55.7 VCCCN49I VVDS:  VCC6680 VVSAM: N/A     WRITR: VCC6680 XHSM:  N/A     XSMS:  VCC6680 XSSI:  VCC6680

18:24:55.7 VCCCN03I DASD Space Collector initialization completed



Figure 3-1.  VCC Initialization and Parameter Message Log
```

Figure 3-1. VCC Initialization and Parameter Message Log

```
C CA MICS                                          Space Collector                 Version 3.2.1 @ VCC6690
 VTOC Scan Log: DDNAME(VCCVTMSG)                                                  Copyright (C) 2013 CA. All Rights Reserved

 17:23:33.8 VCCVT25I TASK(02) VTOC collector initialization processing is complete; TCB(0079A088)
 17:23:33.8 VCCVT10I TASK(02) VTOC scan initiation - Vol(MIX009) Unit(2040) Device(3390-9)
 17:23:33.8 VCCVT25I TASK(04) VTOC collector initialization processing is complete; TCB(00796088)
 17:23:33.8 VCCVT10I TASK(04) VTOC scan initiation - Vol(CAT014) Unit(2160) Device(3390-1)
 17:23:33.8 VCCVT25I TASK(01) VTOC collector initialization processing is complete; TCB(0079C088)
 17:23:33.8 VCCVT10I TASK(01) VTOC scan initiation - Vol(SYSD01) Unit(2220) Device(3390-3)
 17:23:33.8 VCCVT25I TASK(03) VTOC collector initialization processing is complete; TCB(007AE2D0)
 17:23:33.8 VCCVT10I TASK(03) VTOC scan initiation - Vol(MVP04Q) Unit(2320) Device(3390-3)
 17:23:33.8 VCCVT25I TASK(07) VTOC collector initialization processing is complete; TCB(007A9088)
 17:23:33.8 VCCVT10I TASK(07) VTOC scan initiation - Vol(R16R02) Unit(2400) Device(3390-9)
 17:23:33.8 VCCVT25I TASK(06) VTOC collector initialization processing is complete; TCB(007A8CF0)

 17:23:33.8 VCCVT10I TASK(06) VTOC scan initiation - Vol(SVC11V) Unit(2500) Device(3390-9)
 17:23:33.8 VCCVT25I TASK(05) VTOC collector initialization processing is complete; TCB(007A8E88)
 17:23:33.8 VCCVT10I TASK(05) VTOC scan initiation - Vol(GSS001) Unit(2620) Device(3390-3)
 17:23:33.8 VCCVT25I TASK(09) VTOC collector initialization processing is complete; TCB(0078B088)
 17:23:33.8 VCCVT10I TASK(09) VTOC scan initiation - Vol(UK0002) Unit(2700) Device(3390-3)
 17:23:33.8 VCCVT10I TASK(18) VTOC scan initiation - Vol(TD5500) Unit(5500) Device(3390-9)
 17:23:34.0 VCCVT11I TASK(08) VTOC scan completion - Vol(VMU061)
 17:23:34.0 VCCVT10I TASK(08) VTOC scan initiation - Vol(DEMS02) Unit(2B41) Device(3390-3)
 17:23:34.0 VCCVT11I TASK(01) VTOC scan completion - Vol(SYSD01)
 17:23:34.0 VCCVT10I TASK(01) VTOC scan initiation - Vol(TSU006) Unit(2240) Device(3390-3)
 17:23:34.2 VCCVT11I TASK(14) VTOC scan completion - Vol(DMX9C6)
 17:23:34.2 VCCVT10I TASK(14) VTOC scan initiation - Vol(OSI005) Unit(2C20) Device(3380-K)
 17:23:34.2 VCCVT11I TASK(06) VTOC scan completion - Vol(SVC11V)
 17:23:34.2 VCCVT10I TASK(06) VTOC scan initiation - Vol(DB3107) Unit(2540) Device(3390-9)
 17:23:34.2 VCCVT11I TASK(03) VTOC scan completion - Vol(MVP04Q)
 17:23:34.2 VCCVT10I TASK(03) VTOC scan initiation - Vol(MVP04G) Unit(2321) Device(3390-3)
 17:23:34.3 VCCVT11I TASK(04) VTOC scan completion - Vol(CAT014)
 17:23:34.3 VCCVT10I TASK(04) VTOC scan initiation - Vol(LOAN31) Unit(2180) Device(3390-3)
 17:23:34.3 VCCVT11I TASK(18) VTOC scan completion - Vol(TD5500)
 17:23:34.8 VCCVT40W TASK(08) Allocation error     - Vol(DEMS02) Ret(04) Rsn(17080002) DSN(IALC78.DB2.COBOL)



Figure 3-2.  VCC VTOC Scan Log
```

Figure 3-2. VCC VT0C Scan Log

```
CA MICS                                          Space Collector                Version 3.2.1 @ VCC6690

Catalog Scan Log: DDNAME(VCCVSMSG)                                              Copyright (C) 2013 CA. All Rights Reserved


17:23:33.7 VCCVC25I VCONTROL Data collector initialization processing is complete; TCB(007C9B60)
17:23:33.7 VCCSC25I TASK(02) VSAM collector initialization processing is complete; TCB(007C9550)
17:23:33.7 VCCSC01I TASK(02) ICF  scan initiation - Vol(MVCA31) DSN(ICF.MASTER.VMVCA31)
17:23:33.7 VCCSC25I TASK(01) VSAM collector initialization processing is complete; TCB(007C96E8)
17:23:33.7 VCCSC25I TASK(03) VSAM collector initialization processing is complete; TCB(007C92C0)
17:23:33.7 VCCSC25I TASK(10) VSAM collector initialization processing is complete; TCB(007AE968)
17:23:33.7 VCCSC25I TASK(09) VSAM collector initialization processing is complete; TCB(007AEBF8)
17:23:33.7 VCCSC25I TASK(06) VSAM collector initialization processing is complete; TCB(007A4088)
17:23:33.7 VCCSC25I TASK(05) VSAM collector initialization processing is complete; TCB(007A6088)
17:23:33.7 VCCSC25I TASK(07) VSAM collector initialization processing is complete; TCB(007A0088)
17:23:33.7 VCCSC25I TASK(08) VSAM collector initialization processing is complete; TCB(007AEE88)
17:23:33.7 VCCSC25I TASK(04) VSAM collector initialization processing is complete; TCB(007C9128)
17:23:33.9 VCCSC01I TASK(08) VVDS scan initiation - Vol(DEV175) DSN(SYS1.VVDS.VDEV175)
17:23:33.9 VCCSC01I TASK(09) VVDS scan initiation - Vol(CAT014) DSN(SYS1.VVDS.VCAT014)
17:23:33.9 VCCSC01I TASK(10) VVDS scan initiation - Vol(WRKD20) DSN(SYS1.VVDS.VWRKD20)
17:23:34.6 VCCSC02I TASK(01) VVDS scan completion - Vol(SVC11V) DSN(SYS1.VVDS.VSVC11V)
17:23:34.6 VCCSC01I TASK(01) ICF  scan initiation - Vol(ACF001) DSN(CATALOG.VACF201) CAT(ICF.MASTER.VMVCA31)
17:23:34.6 VCCSC02I TASK(04) VVDS scan completion - Vol(UK0002) DSN(SYS1.VVDS.VUK0002)
17:23:34.6 VCCSC01I TASK(04) VVDS scan initiation - Vol(DEMS02) DSN(SYS1.VVDS.VDEMS02)
17:23:34.6 VCCSC02I TASK(03) VVDS scan completion - Vol(GSS001) DSN(SYS1.VVDS.VGSS001)
17:23:34.6 VCCSC01I TASK(03) ICF  scan initiation - Vol(FDRCT1) DSN(CATALOG.ABRBASE) CAT(ICF.MASTER.VMVCA31)
17:23:35.0 VCCSC02I TASK(07) VVDS scan completion - Vol(PRD606) DSN(SYS1.VVDS.VPRD606)
17:23:35.0 VCCSC02I TASK(09) VVDS scan completion - Vol(CAT014) DSN(SYS1.VVDS.VCAT014)
17:23:35.0 VCCSC02I TASK(08) VVDS scan completion - Vol(DEV175) DSN(SYS1.VVDS.VDEV175)
17:23:35.1 VCCSC01I TASK(10) VVDS scan initiation - Vol(TSO30Z) DSN(SYS1.VVDS.VTSO30Z)
17:23:35.1 VCCSC01I TASK(07) ICF  scan initiation - Vol(CAT015) DSN(ICF.ACS.USERCAT) CAT(ICF.MASTER.VMVCA31)
17:23:35.1 VCCSC01I TASK(09) ICF  scan initiation - Vol(CAT001) DSN(ICF.ADG.USERCAT) CAT(ICF.MASTER.VMVCA31)
17:23:35.1 VCCSC01I TASK(08) ICF  scan initiation - Vol(ASM008) DSN(ICF.ASM.USERCAT) CAT(ICF.MASTER.VMVCA31)
17:23:35.2 VCCSC02I TASK(05) VVDS scan completion - Vol(DB3069) DSN(SYS1.VVDS.VDB3069)
17:32:15.5 VCCVC02I TASK(05) VSAM collector termination processing is complete; Catalogs(00281) Waits(00004)
17:32:15.6 VCCVC02I TASK(07) VSAM collector termination processing is complete; Catalogs(00212) Waits(00002)
17:32:15.6 VCCVC02I TASK(01) VSAM collector termination processing is complete; Catalogs(00351) Waits(00071)
17:32:15.6 VCCVC02I TASK(03) VSAM collector termination processing is complete; Catalogs(00319) Waits(00007)
17:32:15.6 VCCVC02I TASK(10) VSAM collector termination processing is complete; Catalogs(00319) Waits(00002)
17:32:15.6 VCCVC02I TASK(09) VSAM collector termination processing is complete; Catalogs(00209) Waits(00002)
17:32:15.6 VCCVC02I TASK(06) VSAM collector termination processing is complete; Catalogs(00309) Waits(00003)
17:32:15.6 VCCVC02I TASK(04) VSAM collector termination processing is complete; Catalogs(00329) Waits(00006)
17:32:15.6 VCCVC02I TASK(02) VSAM collector termination processing is complete; Catalogs(00310) Waits(00025)
17:32:15.6 VCCVC02I TASK(08) VSAM collector termination processing is complete; Catalogs(00255) Waits(00002)
17:32:15.6 VCCVC01I VCONTROL Data collection termination processing is complete



Figure 3-3. VCC Catalog Scan Log
```

Figure 3-3. VCC Catalog Scan Log

```
CA MICS                                        Space Collector              Version 3.2.1 @ VCC6690          Page:   1
 Run Status Report: DDNAME(VCCSTATS)                                          Copyright (C) 2013 CA. All Rights Reserved.

 Processing Summary
 ------------------

 SYSID:                                          CA31
 System Name:                                    CA31
 Sysplex Name:                                 PLEXC1
 Hardware Name:                                  MF01
 Sysres Volume:                                MVR1CB
 System Control Program:                     SP7.1.2
 SCP FMID:                                   HBB7770
 SMS Level:                          (z/OS)  01.12.00

 Processor Model Number:        (System z196)  2817

 Processors:                                       10
   CP                                               6
   zAAP                                             0
   zIIP                                             4

 DASD Devices Online:                          6,088
 DASD Devices Scanned:                         6,088

 ICF BCS  Catalogs Scanned:                      689
 ICF VVDS Catalogs Scanned:                    4,201

 VSAM Catalogs Scanned:                            0

 Processing Start Time:    (THU) 2011.174  11:32:04
 Previous Process Time:                       Unknown
 Forced EndTime Enabled:          2011.174  11:17:16

 VTOC Scan Elapsed Time:                     15:38.06
 Catalog Scan Elapsed Time:                  15:38.09

 VCC Records Written:                      3,334,114
   SMS  Configuration                            431
   VTOC Volume                                 5,328
   VTOC Dataset                            1,735,178
   VTOC Multi-Volume                           1,012
   ICF  Catalog                              223,620
   VVDS Catalog                            1,368,545
 VCC Records Bypassed:                             0

 VTOC DSCB Records Read:                   2,715,686
 VTOC DSCB Records Processed:              1,834,755
   SMS Datasets                            1,252,662
   NonSMS Datasets                           508,317
   Dataset Extents                            73,809

 ICF BCS  Catalog Records:                 8,051,625
 ICF BCS  NonVSAM Records:                 5,529,599

 ICF VVDS Catalog Records:                 1,373,117
 ICF VVDS NonVSAM Records:                 1,099,077

 VSAM Catalog Records:                             0

 DASD Device Type Summary
   3380-E                                          2
   3380-K                                         85
   3390-1                                        285
   3390-2                                         97
   3390-3                                      2,470
   3390-9                                      3,077
   3390-A                                         72

 Free Extent Thresholds
   Distribution 01                               100
   Distribution 02                               200
   Distribution 03                               300
```

```
   Distribution 04                            400
   Distribution 05                            600
   Distribution 06                            800
   Distribution 07                          1,000

 PDSE Dataset Statistics
   Detected                                 252,643
     SMS Datasets                           234,865
     NonSMS Datasets                         17,778
   Analyzed                                  67,652
   Excluded                                 184,991

 HFS  Dataset Statistics
   Detected                                  9,351
     SMS Datasets                            6,700
     NonSMS Datasets                         2,651
   Analyzed                                      0
   Excluded                                  9,351

 zSeries File Systems Detected:              4,130

 Uncataloged NonVSAM Datasets:                   ?

 Temporary Datasets Auto-Excluded:          25,801

 Internal Stack HWM:                             2

 Dataset Resolution Queue
   Maximum Length                            2,018

 Extents Resolution Queue
   Maximum Length                            2,018

 Free    Resolution Queue
   Maximum Length                               22

 Message Severity Counts
   Info                                     22,248
   Warning                                   1,451
   Error                                         2
   Critical                                      0

 Writer Pool Statistics

   Message  Element Pool
     Element Limit                             512
     Element HWM                                50
     Element Waits                               0

   Standard Element Pool
     Element Limit                           8,192
     Element HWM                             1,059
     Element Waits                               0

   Extended Element Pool
     Element Limit                             256
     Element HWM                                 7
     Element Waits                               0

   Blocksiz Element Pool
     Element Limit                             128
     Element HWM                                10
     Element Waits                               0




Figure 3-4.  VCC Processing Summary
```

Figure 3-4. VCC Processing Summary

```
CA MICS                                        Space Collector                Version 3.2.1 @ VCC6690           Page:   1
Run Status Report: DDNAME(VCCSTATS)                                           Copyright (C) 2013 CA. All Rights Reserved.

Catalog Scan Summary     (In sort order by VOLUME)
-----------------------------------------------------

                                                          CATALOG ------ CATALOG ENTRY TYPES ------  CATALOG
                                                UNIT CATALOG RECORDS    VSAM           CATALOG          PROCESS
CATALOG NAME                              VOLUME ADDR TYPE       READ CLUSTER     AIX CONNECTOR NONVSAM     TIME PROCESS STATUS
-------------------------------------------------------------------------------------------------------------------------------
ICF.DB2QA.USERCAT                         ACFQA1 25B5 ICF           0       0       0         0       0      .87 Open Error
SYS1.VVDS.VACFQA1                         ACFQA1 25B5 VVDS        173       0       0         0       0      .94
ICF.QADB2.USERCAT                         ACFQA1 25B5 ICF        1610     759       0         5      82      .88
ICF.VACFQA1.USERCAT                       ACFQA1 25B5 ICF        3549     321       4        14    2741     1.76
SYS1.VVDS.VACFQA2                         ACFQA2 527E VVDS          6       0       0         0       0      .67
CATALOG.VACF201                           ACF001 221A ICF        5736    1311       1        67    2920     2.70
SYS1.VVDS.VACF001                         ACF001 221A VVDS        987       0       0         0       0      .58
ICF.TSSFAD                                ACF001 221A ICF           0       0       0         0       0     1.52 Open Error
SYS1.VVDS.VADG001                         ADG001 2111 VVDS        289       0       0         0     202      .46
SYS1.VVDS.VADG002                         ADG002 2984 VVDS        251       0       0         0     192      .59
SYS1.VVDS.VADG003                         ADG003 2985 VVDS        295       0       0         0     189      .94
SYS1.VVDS.VADG004                         ADG004 2986 VVDS        302       0       0         0     200      .91
SYS1.VVDS.VADG005                         ADG005 298C VVDS        308       0       0         0     200     1.88
SYS1.VVDS.VADG006                         ADG006 202A VVDS        313       0       0         0     199      .82
SYS1.VVDS.VADG007                         ADG007 298A VVDS        308       0       0         0     194     1.27
SYS1.VVDS.VADG008                         ADG008 298B VVDS        327       0       0         0     216      .66
SYS1.VVDS.VADG009                         ADG009 2113 VVDS        296       0       0         0     203      .34
SYS1.VVDS.VADG010                         ADG010 2012 VVDS        280       0       0         0     187      .51
SYS1.VVDS.VADG011                         ADG011 299A VVDS        277       0       0         0     204     1.19
SYS1.VVDS.VADG012                         ADG012 2D70 VVDS        270       0       0         0     206     1.12
SYS1.VVDS.VADG013                         ADG013 2D71 VVDS        224       0       0         0     161     1.49
SYS1.VVDS.VADG014                         ADG014 2D72 VVDS        294       0       0         0     209      .85
SYS1.VVDS.VADG015                         ADG015 2D73 VVDS        252       0       0         0     179      .49
SYS1.VVDS.VADG016                         ADG016 2D74 VVDS        254       0       0         0     173     1.04
SYS1.VVDS.VADG017                         ADG017 2D87 VVDS        275       0       0         0     198      .76
SYS1.VVDS.VANLS02                         ANLS02 2822 VVDS         56       0       0         0      54      .78
SYS1.VVDS.VANL001                         ANL001 281C VVDS         13       0       0         0       0      .98
SYS1.VVDS.VANL002                         ANL002 2820 VVDS          7       0       0         0       0     1.01
ICF.APCD.USERCAT                          APCD06 23D2 ICF       37441     569       1       319   35371     4.81
SYS1.VVDS.VAPCD06                         APCD06 23D2 VVDS        21       0       0         0       0     1.10
SYS1.VVDS.VAPCD07                         APCD07 23D3 VVDS        30       0       0         0       0     1.07
SYS1.VVDS.VAPCD08                         APCD08 23D4 VVDS        72       0       0         0       0     1.05
SYS1.VVDS.VAPCD09                         APCD09 23D5 VVDS        26       0       0         0       0     1.08
SYS1.VVDS.VAPCD10                         APCD10 23D6 VVDS        21       0       0         0       0     1.07
SYS1.VVDS.VAPCD11                         APCD11 2899 VVDS        16       0       0         0       0      .69
SYS1.VVDS.VAPCD12                         APCD12 2604 VVDS        36       0       0         0       0     1.07
SYS1.VVDS.VAPCD13                         APCD13 21CF VVDS        20       0       0         0       0      .83
SYS1.VVDS.VAPCD14                         APCD14 23BC VVDS        12       0       0         0       0     1.06
SYS1.VVDS.VAPCD15                         APCD15 23BD VVDS        60       0       0         0       0     1.11
SYS1.VVDS.VAPCD16                         APCD16 5170 VVDS        31       0       0         0       0      .82

----- End of Table -----
```

$FIGC Figure 3-5.  VCC Catalog Scan Summary (Part 1 of 2)

```
CA MICS                                          Space Collector                Version 3.2.1 @ VCC6690          Page:   2
Run Status Report: DDNAME(VCCSTATS)                                             Copyright (C) 2013 CA. All Rights Reserved.

Catalog Scan Summary     (In sort order by VOLUME)
-----------------------------------------------------

Processing Record Totals (Catalog Scan)

  Catalog Records                       9,234,883
  Cluster Records                         242,474
  AIX Records                               1,393
  Connector Records                        59,618
  Non-VSAM Records                      6,400,870

Processing Status Totals (Catalog Scan)

  Normal Completion                         4,320
  Allocation Error                              6
  Locate Error                                  0
  MODCB Error                                   0
  Not Processed                                 0
  Open Error                                  189
  Point Error                                   0
  Processing Error                              0
  Read Error                                    2
  Scan Abend(s)                                 0
  SMF Write Error                               0
  Unknown Status                                0
  VSAMCAT D/I SOF                               0
  VSAMCAT EXT SOF                               0
```

$FIGC Figure 3-5.  VCC Catalog Scan Summary (Part 2 of 2)

```
CA MICS                                        Space Collector                Version 3.2.1 @ VCC6690          Page:   1
 Run Status Report: DDNAME(VCCSTATS)                                            Copyright (C) 2013 CA. All Rights Reserved.

 DFSMS Storage Group Summary
 ---------------------------

 STORAGE                       TOTAL SPACE PERCENT  VOLUME
 GROUP NAME ---- STATUS ----      (TRACKS)    USED   COUNT ---------------------------- VOLUME LIST -----------------------------
 ADGVOLS    ENABLED              2,555,100    65.7      17 ADG001,ADG002,ADG003,ADG004,ADG005,ADG006,ADG007,ADG008,ADG009,ADG010,
                                                           ADG011,ADG012,ADG013,ADG014,ADG015,ADG016,ADG017.
 AUTMVOLS   ENABLED                300,600    74.2       6 AUTM01,AUTM02,AUTM03,AUTM04,AUTM05,AUTM06.
 DEMOED     ENABLED              1,202,400    65.8      12 DEMS01,DEMS02,DEMS03,DEMS04,DEMS05,DEMS06,DEMS07,DEMS08,DEMS09,DEMS10,
                                                           DEMS11,DEMS12.
 DISKEDUC   ENABLED                  1,710     7.7       1 DEMD01.
 DUMPVOLS   ENABLED              6,596,100    63.2      28 DMP010,DMP011,DMP012,DMP013,DMP014,DMP015,DMP016,DMP017,DMP018,DMP019,
                                                           DMP020,DMP021,DMP022,DMP023,DMP024,DMP025,DMP026,DMP027,DMP028,DMP029,
                                                           DMP030,DMP031,DMP032,DMP033,DMP034,DMP035,DMP036,DMP037.
 DUMPVOLS   DISABLED NEW           150,300    87.9       1 DMP022.
 GMMPPD     ENABLED                 50,100    37.7       1 MMPP01.
 HFSMINIS   ENABLED              1,119,360    88.8      11 HFSMNA,HFSMN1,HFSMN2,HFSMN3,HFSMN4,HFSMN5,HFSMN6,HFSMN7,HFSMN8,HFSMN9,
                                                           HFS240.
 IDIVOLS    ENABLED                601,200    90.1       8 IDI001,IDI002,IDI003,IDI004,IDI005,IDI006,IDI007,IDI008.

 ----- End of Table -----

 (Totals for All Groups)      328,107,555    46.9   2,142



Figure 3-6.  VCC DFSMS Storage Group Summary
```

Figure 3-6. VCC Storage Group Summary

```
CA MICS                                        Space Collector                Version 3.2.1 @ VCC6690          Page:   1
 Run Status Report: DDNAME(VCCSTATS)                                          Copyright (C) 2013 CA. All Rights Reserved.

 VTOC Scan Summary        (In sort order by VOLUME)
 ------------------------------------------------

                       SCAN %TOTAL TOTAL ---------------- DSCB USE STATISTICS ---------------- V V I S S E
 VOLUME UNIT DEVICE  ELAPSED  SPACE  VTOC             ----------- FORMAT TYPE -----------  HIGH I V C M H A STORAGE
        ADDR  TYPE      TIME   USED  TRKS TOTAL  USED    F1    F3    F5    F7    F8    F9  USED X D F S R V GROUP    PROCESS STATUS
 ---------------------------------------------------------------------------------------------------------------------------------
 DMP035 20F6 3390-9     .010   67.6   150  7500    29    26     1     1     0     0     0    38 Y Y N Y Y N
 DMP036 20F7 3390-9     .011   55.1   449 22450    26    22     2     1     0     0     0    39 Y Y N Y Y N
 DMP037 20F8 3390-9     .013   57.7   449 22450    32    29     1     1     0     0     0    52 Y Y N Y Y N
 DMP038 20F9 3390-9     .010   87.5   449 22450    28    24     2     1     0     0     0    42 Y Y N Y Y N
 EAV002 2200 3390-A     .876   29.3   449 22450   334    76    64     1     0    96    96   451 Y Y N N Y Y
 GSS001 0FAF 3390-3    7.399    9.0    45  2250    49    47     0     1     0     0     0   227 Y Y Y N Y N
 HFSMNA 20A0 3390-9     .058   88.4    44  2200    16    14     0     1     0     0     0    17 Y Y N Y Y N
 HFSMNB 050F 3390-9     .072   94.2   449 22450    16    14     0     1     0     0     0    17 Y Y N Y Y N
 HFSMNC 210D 3390-9     .009   87.4   449 22450    13    11     0     1     0     0     0    13 Y Y N Y Y N
 HFSMND 2128 3390-9     .010   28.1   150  7500     8     6     0     1     0     0     0     9 Y Y N Y Y N
 HFSMNE 2129 3390-9     .012   60.1   150  7500    12    10     0     1     0     0     0    13 Y Y N Y Y N
 HFSMNF 2066 3390-9     .010   61.2   150  7500    10     8     0     1     0     0     0    10 Y Y N Y Y N
 HFSMN0 2139 3390-9     .035   57.7   150  7500    10     8     0     1     0     0     0    10 Y Y N Y Y N
 HFSMN1 2832 3390-3     .021   92.7    14   700    22    18     2     1     0     0     0    68 Y Y N Y Y N
 HFSMN2 2168 3390-3     .035   97.0    45  2250    28    26     0     1     0     0     0    29 Y Y N Y Y N
 HFSMN3 2663 3390-3     .013   90.2    45  2250    21    19     0     1     0     0     0    22 Y Y N Y Y N
 HFSMN4 29E5 3390-3     .026   89.4    45  2250    24    22     0     1     0     0     0    24 Y Y N Y Y N
 HFSMN5 2404 3390-9     .009   93.6    44  2200    15    13     0     1     0     0     0    16 Y Y N Y Y N
 HFSMN6 22DA 3390-9     .009   90.1    44  2200    31    29     0     1     0     0     0    31 Y Y N Y Y N
 HFSMN7 25FA 3390-9     .022   88.1   449 22450    26    24     0     1     0     0     0    26 Y Y N Y Y N
 HFSMN8 2420 3390-9     .014   94.9   449 22450    23    21     0     1     0     0     0    24 Y Y N Y Y N
 HFSMN9 2421 3390-9     .023   99.2   449 22450    13    11     0     1     0     0     0    16 Y Y N Y Y N
 HFS240 21F8 3390-2     .014   47.9    14   700    28    26     0     1     0     0     0    28 N Y N Y Y N
 IBM001 2078 3390-9    6.639   82.0    45  2250   481   464    15     1     0     0     0   526 Y Y N Y Y N
 IBM002 272A 3390-9    8.790   33.1    45  2250   489   478     9     1     0     0     0   497 Y Y N Y Y N
 IBM003 2260 3390-9    4.170   63.9    45  2250   358   345    11     1     0     0     0   374 Y Y N Y Y N
 IBM004 2D9B 3390-9    7.680   49.8    45  2250   365   353    10     1     0     0     0   366 Y Y N Y Y N
 IBM005 2D3E 3390-9    3.622   91.8    45  2250   270   259     9     1     0     0     0   270 Y Y N Y Y N
 IBM006 228B 3390-9    4.038   87.5    44  2200   307   299     6     1     0     0     0   307 Y Y N Y Y N
 IBM007 2151 3390-9    9.495   98.6   150  7500   249   233    14     1     0     0     0   412 Y Y N Y Y N
 IBM008 292F 3390-9    8.635   93.2   150  7500   194   182    10     1     0     0     0   194 Y Y N Y Y N
 IBM009 28CC 3390-9    3.841   98.5   449 22450   147   137     8     1     0     0     0   155 Y Y N Y Y N
 IBM010 2109 3390-9    5.532   96.9   449 22450   202   190    10     1     0     0     0   233 Y Y N Y N N
 IBM011 20FA 3390-9    4.329   98.1   449 22450   236   221    13     1     0     0     0   246 Y Y N Y Y N
 IBM012 2D84 3390-9    5.270   97.8  1214 60700   217   208     7     1     0     0     0   253 Y Y N Y N N
 IBM013 20FE 3390-9    2.874   98.0   449 22450    61    58     1     1     0     0     0    71 Y Y N Y Y N
 IBM014 20FF 3390-9    5.773   98.7   449 22450    79    67    10     1     0     0     0   113 Y Y N Y Y N
 IBM015 2102 3390-9    3.141   97.7   449 22450   109   103     4     1     0     0     0   123 Y Y N Y Y N
 IBM016 2103 3390-9    3.749   95.2   449 22450   109   104     2     1     1     0     0   109 N Y N Y Y N

 ----- End of Table -----
```

$FIGC Figure 3-7.  VCC VTOC Scan Summary (Part 1 of 2)

```
CA MICS                                        Space Collector          Version 3.2.1 @ VCC6690          Page:   2
Run Status Report: DDNAME(VCCSTATS)                                     Copyright (C) 2013 CA. All Rights Reserved.

VTOC Scan Summary        (In sort order by VOLUME)
-------------------------------------------------

Processing Detail Totals (VTOC Scan)

  Scan Time Minimum (HD032A)                0.005
  Scan Time Maximum (MOTM10)            26:02.321
  Scan Time Average                        11.403

  DASD Total Volumes                        5,507
  DASD Total Tracks                   643,017,465
  DASD Total Allocated                275,455,330
  DASD Percent Used                          42.8%

  VTOC Total Tracks                     1,000,508
  VTOC Total DSCB                      50,027,104
  VTOC Total DSCB Used                  1,251,404

  VTOC Format 1 DSCBs                   1,194,911
  VTOC Format 3 DSCBs                      45,881
  VTOC Format 5 DSCBs                       5,011
  VTOC Format 7 DSCBs                         652
  VTOC Format 8 DSCBs                          71
  VTOC Format 9 DSCBs                          71


Processing Status Totals (VTOC Scan)

  Normal Completion                         4,710
  Device Not Ready                              0
  DSN Residuals                                 0
  EXT Not Found                                 0
  EXT Residuals                                 1
  Incomplete Scan                               0
  Lost Tracks                                  36
  Migration Lvl 1                               0
  Mount Attributes                              0
  No VTOC Found                                 0
  Processing Error                              0
  PDSE Extended                                 0
  PDSE No Extended                             11
  Record Overflow                               0
  Reserve Bypassed                              0
  Scan Abend(s)                                 0
  Status Bypassed                               0
  Volume Excluded                               0
  Volume Not Read                               0
  Volume Reserved                               0
  VM Volume                                   749
```

$FIGC Figure 3-7.  VCC VTOC Scan Summary (Part 2 of 2)

```
CA  CA MICS                       Space Collector         Version 3.2.1 @ VCC6690                Page:   1
 Copyright (C) 2013 CA. All Rights Reserved.

 HSM Status Report: DDNAME(VCCHSMSG)                        DFHSM 1.5.9

 VCCHS02I Previous Data Collection (MIGRATE Data):              Unknown
 VCCHS02I Previous Data Collection (BACKUP  Data):              Unknown

 VCCHS03I Starting Data Collection:                  2007.236  17:23:33
 VCCHS04I Ending   Data Collection:                  2007.236  17:25:12

 VCCHS05I Elapsed Processing Time:                              00:01:39

 VCCHS06I Number of Type "M" Records in --ML2-- Read:           622,635
 VCCHS06I Number of Type "M" Records in --ML1-- Read:             1,086
 VCCHS06I Number of Type "M" Records in -SDSP-- Read:                 0

 VCCHS07I Number of HSM -- MIGRATE -- Records Written:          623,721
 VCCHS07I Number of HSM -- BACKUP  -- Records Written:                0
 VCCHS07I Number of HSM -- DASD-CP -- Records Written:            8,213
 VCCHS07I Number of HSM -- TAPE-CP -- Records Written:                3

 VCCHS28I Number of type "M" records on ML2 skipped:                  0
 VCCHS23I Number of records skipped (Invalid LRECL):                  0

 VCCHS08I Number of Records Written:                            631,938




Figure 3-8.  VCC HSM Status Report
```

Figure 3-8. VCC HSM Status Report

```
USS File System Collector        -  Version 3.2.1 @ VCC6550    SYSID: CA31          01 SEP yyyy   Page:     1
Copyright (C) 2009 CA. All Rights Reserved.

Message Log: DDNAME(SYSPRINT)

17:08:08.508 00 VCC0012I USS File System Collector Initialization In Progress
17:08:08.540 00 VCC0031I Input parameter file VCCPARMS   on volume *JES*
17:08:08.540 00 VCC0002I Parameters specified via VCCPARMS DD statement:
17:08:08.540 00 VCC0002I >>    DATALEVEL=(ALL)
17:08:08.540 00 VCC0002I >>    HFSSUBT=05
17:08:08.541 00 VCC0070I Control Data Set MDI.VCC6490.TST.UFSCNTL is on volume TSU034
17:08:09.055 00 VCC0078I Control Data Set MDI.VCC6490.TST.UFSCNTL has been initialized
17:08:09.095 00 VCC0020I Set VolumeTableLength(1034K)
17:08:09.119 00 VCC0021I 114 Mounted file systems found
17:09:01.027 00 VCC0013I USS File System Collector Initialization Complete
17:09:01.058 02 VCC0017I Read USS File System Subtask 02 Started
17:09:01.058 01 VCC0017I Read USS File System Subtask 01 Started
17:09:01.059 05 VCC0017I Read USS File System Subtask 05 Started
17:09:01.059 04 VCC0017I Read USS File System Subtask 04 Started
17:09:01.059 03 VCC0017I Read USS File System Subtask 03 Started
17:09:01.059 02 VCC0018I Now scanning file system OMVS.CASPOOL.HFS
17:09:01.060 01 VCC0018I Now scanning file system OMVS.CA31.CTG402.HFS
17:09:01.061 05 VCC0018I Now scanning file system OMVS.CA31.CTG500.HFS
17:09:01.061 04 VCC0018I Now scanning file system OMVS.CA31.DEV
17:09:27.122 03 VCC0018I Now scanning file system TEMP
17:09:27.124 03 VCC0017I Read USS File System Subtask 03 Ended
17:09:27.124 03 VCC0100I   File Systems Processed...........16
17:09:28.702 02 VCC0017I Read USS File System Subtask 02 Ended
17:09:28.702 02 VCC0100I   File Systems Processed...........34
17:09:49.675 04 VCC0017I Read USS File System Subtask 04 Ended
17:09:49.675 04 VCC0100I   File Systems Processed...........20
17:10:16.607 01 VCC0017I Read USS File System Subtask 01 Ended
17:10:16.607 01 VCC0100I   File Systems Processed...........26
17:11:25.731 05 VCC0017I Read USS File System Subtask 05 Ended
17:11:25.731 05 VCC0100I   File Systems Processed...........14
17:16:27.609 36 VCC0014I 544,908 records written to MDI.VCC6490.TST.UFSDATA on volume DMP015
17:16:32.643 00 VCC0010I Processing completed, Return Code 0




Figure 3-9.  USS File System Message Log
```

Figure 3-9. USS File System Message Log

```
USS File System Collector        -  Version 3.2.1 @ VCC6550    SYSID: CA31                          01 SEP yyyy   Page:     1
Copyright (C) 2009 CA. All Rights Reserved.

Run Statistics: DDNAME(VCCSTATS)


Processing Summary
------------------

System Name:                                          CA31
SMFID:                                                CA31
Operating System Level:                    SP7.0.8   HBB7730
SMS or DFP Level:                                  01.08.00
Central Processors:                                       8
Total Online DASD Devices:                            4,541
-------------------------------------------------------------
USS File Systems Detected:                              123
USS File Systems Monitored:                             116
USS File System Files Detected:                   2,165,491
USS File System Directories:                        548,925
-------------------------------------------------------------
Previous Collection Run On System:                    CA31
Previous Collection Date:            03/01/2008 17:08:08.541
Collector Run Number:                                 0042
Collection Duration:                                8:23.637
Collection CPU Time:                                 138.546
-------------------------------------------------------------
Collector Records Written                           544,908
Collector Subtasks Started:                              16
Collector Subtask Buffer Wait Count:                      0
-------------------------------------------------------------
Collection Warning Messages Issued:                       0
Collection Error Messages Issued:                         0
Collection Highest Completion Code:                       0
-------------------------------------------------------------
```

Figure 3-10.  USS File System Processing Summary

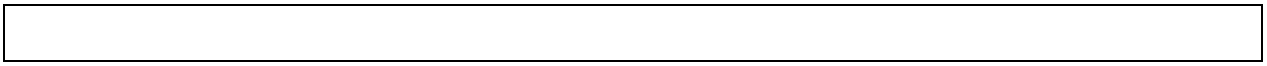


Figure 3-10. USS File System Processing Summary

```
USS File System Collector        -  Version 3.2.1 @ VCC6550    SYSID: CA31         01 SEP yyyy   Page:     2
Copyright (C) 2009 CA. All Rights Reserved.

Run Statistics: DDNAME(VCCSTATS)

USS File Systems Summary
------------------------

                                  T VOLUME DEVICE UNIT    ALLOC   %USED  FILE COUNT
-------- FILE SYSTEM NAME --------- - ------ -TYPE- ADDR --TRACKS- ------ ----------- --STATUS--- ----------- MOUNT POINT -----
OMVSGRP.OMLVL1.CA31.CASPOOL.HFS     H APCM06 3390-3 26DB       675  86.81         392 ACT R/W SCN /u/users/omlvl1
OMVSGRP.OMQA.CA31.CASPOOL.HFS       H APCM06 3390-3 26DB     1,860  98.54         880 ACT R/W SCN /u/users/omqa
OMVSGRP.KUOCH01.TSO44.HFS           H APCM07 3390-3 2A9C       750   5.73         105 ACT R/W SCN /u/users/kuoch01
OMVSGRP.OPSMVS.HFS                  H AUTM06 3390-3 2AAF     1,005  87.26       1,971 ACT R/W SCN /u/users/opsmvs
OMVSSYS.DB2910.GA.CA31.PDO0712.HFS  H DB2CAC 3390-9 2091     1,500  47.73         334 ACT R/W SCN /sys/DB2V910
OMVSSYS.DB2810.CA31.PDO0614C.HFS    H DB2CA8 3390-3 2261     4,500  18.95         638 ACT R/W SCN /sys/DB2810
OMVSGRP.USER.DCMDEV.CA31.HFS        H DCMQAF 3390-9 2100    12,150  89.46      16,127 ACT R/W SCN /u/users/datacom
OMVSGRP.USER.DCMDEV.HFS             H DCMSP2 3390-9 2437    13,350  76.26       7,918 ACT R/W SCN /u/users/dcmdev
OMVSSYS.JAVATM2.V1R4M0.BASE2.HFS    H IBM003 3390-3 21B9     3,600  88.19       1,627 ACT R/O SCN /sys/javatm2/v1r4m0
OMVSSYS.WSED.V6.V500.HFS            H IBM003 3390-3 21B9     2,025  95.55         260 ACT R/W SCN /sys/wsed6
OMVSSYS.GNUMAKE.V39R1M1.HFS         H IBM005 3390-3 2A97        45  82.22         174 ACT R/O SCN /sys/gnumake
OMVSSYS.PERL.V5R6M1.HFS             H IBM005 3390-3 2A97       615  97.39       1,270 ACT R/O SCN /sys/perl
OMVSSYS.WAS51.CONFIG.B51A.HFS       H IBM005 3390-3 2A97     8,250  72.46      50,970 ACT R/W SCN /sys/wascfgB51A
OMVS.MVSSYS.WEBSPHMQ.V6R0M0.ETP.HFS H IBM007 3390-9 2027       300  78.00         339 ACT R/O SCN /sys/WASMQ600
OMVSSYS.JAVA64BT.V5R0M0.SR4.HFS     H IBM007 3390-9 2027     5,625  90.48       2,140 ACT R/O SCN /sys/java64bt/v5r0m0
OMVSSYS.WAS60.CONFIG.B60A.HFS       H IBM009 3390-9 20CF     2,175  63.54      43,760 ACT R/W SCN /sys/wascfgB60A
OMVSSYS.WEBSPHER.V5R1M0.BASE.SBBOHF H IBM009 3390-9 20CF    13,650  94.46      11,539 ACT R/O SCN /sys/websph51/bbojava
OMVSSYS.JAVATM2.V1R4M1.SR2A.HFS     H IBM010 3390-9 2109     3,600  90.97       1,629 ACT R/O SCN /sys/javatm2/v1r4m1
OMVSSYS.WEBSPHER.V5R1M0.W510215.SBB H IBM011 3390-9 E100    13,650  98.90      11,941 ACT R/O SCN /sys/websph51B/bbojava
OMVSSYS.JAVA64BT.V1R4M2.SR7.HFS     H IBM013 3390-9 2E2B     4,500  87.97       1,588 ACT R/O SCN /sys/java64bt/v1r4m2
OMVSSYS.WAS.V6R0M29.UK13441.SBBOHFS H IBM013 3390-9 2E2B    43,575  96.58      62,784 ACT R/O SCN /sys/websph60
OMVSSYS.WAS.V6R0M29.UK13441.SBBOHFS H IBM013 3390-9 2E2B    43,065  97.81      62,814 ACT R/O SCN /sys/websph60sitem
TEMP                                T V@CA31 V3380  VIRT     3,277   1.40          29 ACT R/W SCN /SYSTEM/tmp

- End of Report -
```

Figure 3-11.  USS File Systems Summary

Figure 3-11. USS File Systems Summary

# 3.2 ASTEX Migration Manager and CA MICS

The ASTEX Migration Manager is an optional component of ASTEX that was released in GA status on July 11, 1994. It reduces overall CPU and I/O requirements of HSM by selecting data sets for migration based on historical access patterns and probability of future access.

Previously CA-ASTEX could trigger the daily collection of VTOC, VVDS, and HSM data instead of having an additional collection for CA MICS. This is no longer the case.  You must run a separate collection for ASTEX Migration Manager and for CA MICS.

# Chapter 4: INSTALLATION

This chapter presents a general overview of the installation process for the VCC product and lists the detailed tasks necessary for a successful installation. All of the material should be reviewed carefully BEFORE any installation steps are performed.

This section contains the following topics:



## 4.1 Installation Overview

CA MICS Space Collector Option (VCC) installation process consists of unloading the component distribution tape, allocating the VCC data sets, making any decisions that require run-time parameter specification, creating the run-time parameter data set, and copying the VCC modules into an APF-authorized library. This process is detailed in the sections that follow.

### 4.1.1 Distribution Materials

If VCC is to be installed at a host site that has the CA MICS product, VCC is distributed on the complete CA MICS distribution tape and is loaded into the CA MICS libraries before you enter this checklist.

If VCC is to be installed at a remote site where CA MICS itself is not installed, VCC is distributed on a magnetic tape containing 22 files that are in IBM IEBCOPY-unload format. This tape is a standard label tape (SL format) with a volume serial and density marked on the external label.

The VCC modules, JCL, and this guide are stored in five of the 22 files on the tape. If you are only installing VCC, the remaining files will contain dummy entries and can be ignored.

are as follows:

| File | Data Set | Description |
|---|---|---|
| 2 | CAI.MICS.ASM | VCC Assembler source code. Sample User Exits are provided here. If a CA MICS system or component installation, will also include CA MICS Assembler source. |
| 4 | CAI.MICS.CNTL | VCC allocation, run, and install JCL. If a CA MICS system or component installation, will also include CA MICS JCL. |
| 6 | CAI.MICS.DOC. TEXT | CA MICS Space Collector Option Guide. If a CA MICS system or component installation, will also include CA MICS text. |
| 11 | CAI.MICS.LOAD | VCC load modules. If a CA MICS system or component installation, will also include CA MICS load modules. |
| 12 | CAI.MICS.MACLIB | VCC Assembler macros. If a CA MICS system or component installation, will also include CA MICS Assembler macros. |

#### 4.1.1.1 Installation Checklist

This checklist is ONLY for completing the installation of VCC on a host system where CA MICS is already installed.

Installation Checklist

____ 1. Copy the VCC load modules to an authorized library.

This step involves copying the VCC load modules into an APF-authorized library. This can be done using one of the following methods:

- Use VCCCOPY in sharedprefix.MICS.CNTL. This invokes IEBCOPY with a complete list of VCC modules. The BLKSIZE of the APF-authorized library must be at least as large as the BLKSIZE of sharedprefix.MICS.LOAD.

- Use the SAS PDSCOPY facility to copy the VCC load modules to an authorized library. PDSCOPY will do any reblocking that is necessary. VCCCOPY contains the complete list of modules.

  IMPORTANT: If you create your own link-edit to move the VCC load modules, you must remember to make sure that these modules are:

  1. Link-edited with the RENT,REUS attributes. If the VCCGWA and VCCGWB modules are being link-edited, then they must be done so without the RENT,REUS attributes.

  2. Link-edited specifying SETCODE AC(1) for VCC module VCCNTRL.

  3. Link-edited specifying SETCODE AC(1) for VCC module VCCMAIN.

____ 2. Create the VCC run-time parameter data set.

We suggest that the VCC run-time parameters be kept in the sharedprefix.MICS.PARMS library. Refer to Chapter 3 for details on VCC parameter specification before you attempt to create the VCC parameter data set. The parameters specify the DASD devices and VSAM catalogs that are to be scanned. Other specifications include:

- SMF record number(s)
- VCC recording medium (SMF or to a data set)

- VCC tuning parameters
- Inclusion or exclusion of data sets

____ 3. Allocate the VCC operational data sets.

If the number of DASD volumes and data sets VCC will be recording is small, you may use SMF recording. Otherwise, due to the speed at which VCC runs and the amount of data generated at larger sites, it is essential that you record VCC data in a data set other than SMF. This will eliminate the problem of the z/OS SMF data sets filling up quickly and causing system-wide degradation due to SMF buffer shortages.

Use member VCCALLOC in sharedprefix.MICS.CNTL to allocate the VCC operational data sets.

VCC requires the VCCHSM data set only if HSM=Y is coded in VCC.PARMS. If you are writing HSM data directly to the VCCHSM data set, you should allocate a large data set as shown in the example below. If you are using SMF as the recording source for HSM data, the file will be used as a work file and you need to allocate only a few tracks. If the data set is omitted, VCC will not process HSM data and issue a return code 8 for this step.

See Chapter 2, Planning, for information regarding the amount of data generated by VCC. This will help you determine the correct space assignments for the VCC.SMFDATA and VCC.HSMDATA data sets.

____ 4. Set up the VCC operational JCL.

Members VCCRUN and VCCRUNH in library sharedprefix.MICS.CNTL contain sample VCC operational JCL. Use this JCL as a model for constructing your VCC job streams.

VCC is now installed and ready to run. Refer to Chapter 3, Usage Guidelines, to determine the criteria for when to run VCC and how to determine what data to collect.

#### 4.1.1.2 Copying VCC Modules to Remote Sites

If you want to run VCC at a remote site, you can copy the VCC load modules to that site. An example of the JCL is provided in library sharedprefix.MICS.CNTL, member VCCCOPY.

### 4.1.2 Special Installation Considerations

The following items should be examined prior to running the VCC:

- The VCC, by default, uses SMF record number 198. If a current user SMF record number conflicts with this number, a new unique number must be explicitly assigned in the VCC run-time options.

- If you are installing the CA MICS VTOC/Catalog Activity component (VCA), the new SMF record number must also be changed in CA MICS. Specific details may be found in the CA MICS installation instructions.

### 4.1.3 Installation Details

The VCC installation process is not difficult when compared with other software products. Because the VCC comes preassembled and link-edited, no assembly is required. SMP support is not necessary because the VCC makes no modifications to the operating system.

The following sections provide detailed installation checklists for VCC:

1 - Installation Checklist for CA MICS Sites
(VCC Installation with CA MICS base installed)
2 - Installation Checklist for Non-CA MICS Sites
(VCC Installation without CA MICS base installed)

### 4.1.3.1 Installation Checklist for CA MICS Sites

This checklist is ONLY for completing the installation of VCC on a host system where CA MICS is already installed. If you are installing VCC in a standalone environment, outside of the CA MICS system, follow the checklist in Section 4.1.3.2.

Installation Checklist

____ 1. Copy the VCC load modules to an authorized library.

This step involves copying the VCC load modules into an APF-authorized library. The recommended method is this:

- Use VCCCOPY in sharedprefix.MICS.CNTL. This invokes IEBCOPY with a complete list of VCC modules. The BLKSIZE of the APF-authorized library must be at least as large as the BLKSIZE of sharedprefix.MICS.LOAD.

Alternatively, you could do this:

- Use the SAS PDSCOPY facility to copy the VCC load modules to an authorized library. PDSCOPY will do any reblocking that is necessary. VCCCOPY contains the complete list of modules.

____ 2. Create the VCC run-time parameter data set.

We suggest that the VCC run-time parameters be kept in the sharedprefix.MICS.PARMS library. Refer to Chapter 3 for details on VCC parameter specifications before you attempt to create the VCC parameter data set. The parameters specify the DASD devices and VSAM catalogs that are to be scanned. Other specifications include:

- SMF record number(s)
- VCC recording medium (SMF or to a data set)
- VCC tuning parameters
- Inclusion or exclusion of data sets

____ 3. Allocate the VCC operational data sets.

If the number of DASD volumes and data sets VCC will be recording is small, you may use SMF recording. Otherwise, due to the speed at which VCC runs and the amount of data generated at larger data centers, it is essential that you record VCC data in a data set other than SMF. This will

```
eliminate the problem of the SMF MAN files filling up quickly
and causing system-wide degradation due to SMF buffer
shortages.

Use sample JCL member VCCALLOC in sharedprefix.MICS.CNTL as a
model in allocating the VCC operational data sets.

See Chapter 2 of this guide, Planning, for information
regarding the amount of data generated by VCC.  This will
help you determine the correct space assignments for the
VCCDATA and VCCHSM data sets.


____  4.  Set up the VCC operational JCL.

Members VCCRUN and VCCRUNH in library sharedprefix.MICS.CNTL
contain sample VCC operational JCL.  Use these
members as models in constructing your VCC job streams.

VCC is now installed and ready to run.  Refer to Chapter 3,
Usage Guidelines, to determine the criteria for when to run
VCC and how to determine what data to collect.
```

### 4.1.3.2 Installation Checklist for non-CA MICS Sites

VCC can be installed at non-CA MICS (remote) sites. Data collected by VCC running at the remote site must be transported to the host for processing by VCA.

VCC is a separately licensed product and you are required to pay a license fee for each site that runs it. For further information, contact your CA sales representative.

To install VCC at a remote site, you must build a VCC remote tape at the host site and then install it at the remote site. After installation is complete, you must arrange to have the data transported to your host site for VCA processing.

Complete Steps 1 through 4 to build a VCC remote tape at your host site.

___ 1. Prepare the generation job.

Edit sharedprefix.MICS.CNTL(VCCBLD). Add a job card and update the data set names with your sharedprefix.

___ 2. Generate the distribution tape.

Run sharedprefix.MICS.CNTL(VCCBLD). Ensure that the job completes with a condition code of zero.

___ 3. Print the VCC documentation.

Use the CA MICS Workstation Facility (MWF) to print a copy of the CA MICS Space Collector Option Guide or download and print the PDF from http://ca.com/support.

___ 4. Ship the tape and the VCC guide to the remote site.

Complete the following steps at the remote site.

___ 5. Create the following JCL at the remote site. It will load the remote VCC tape created at your host site.

```
//jobname  JOB  ...
//COPY     EXEC PGM=IEBCOPY,REGION=512K
//SYSPRINT DD   SYSOUT=*
```

```
//SYSUT3   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4   DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//#CNTL    DD   DSN=CAI.MICS.LIBLOAD.CNTL,
//              DISP=SHR,DCB=DEN=4,
//              UNIT=____,VOL=SER=______,
//              LABEL=(1,SL,EXPDT=_____)
//CNTL     DD   DISP=SHR,DSN=yourprefix.MICS.CNTL
//SYSIN    DD   *
   C I=#CNTL,O=CNTL
   S M=VCCLOAD
/*

where:

UNIT= is the unit name of the tape device on which the tape
  is to be mounted

VOL=SER= is the volume serial number of the tape

yourprefix.MICS.CNTL is the CNTL library of your choice.
  Note:  If the data set does not exist you must allocate it
  as a PDS with the following DCB attributes:
  RECFM=FB,LRECL=80,BLKSIZE=6160

Submit the job and ensure it completes with a condition code
of zero.


___  6. Load the VCC modules into the installation libraries.

Edit the VCCLOAD member loaded in step 5 and complete
the JCL statements as follows:

//LOAD EXEC LOAD,DSP=NEW,   <-- NEW,LIBRARIES ARE CREATED
//*                         <-- OLD, LIBRARIES MUST EXIST
//      TUNIT='3480',       <-- IS TAPE UNIT OK?
//      TEXPDT=98000,       <-- IS TAPE EXPDT OK?
//      TVOL=______,        <-- REQUIRED --<<
//      LIBUNIT=SYSDA,      <-- IS LIB DASD UNIT OK?
//      LIBVOL=,            <-- DO YOU NEED LIB DASD VOL?
//      WKUNIT=SYSDA,       <-- IS WORK DASD UNIT OK?
//      BLOCK=6160,         <-- LIBRARY BLKSIZE?
//      VPREFIX='_____________', <-- REQUIRED --<<
//      VMICS=                  <-- MICSLEVEL?

where:

DSP= is the disposition of the VCC libraries.
```

```
TUNIT=, TEXPDT=, and TVOL= are the unit, expiration date,
  and volume serial number of the distribution tape.

LIBUNIT= and LIBVOL= are the unit and volume serial numbers
  of the DASD volume for the VCC libraries.

WKUNIT= is the unit for the SYSUT3 and SYSUT4 work data sets
  needed by IEBCOPY.

VPREFIX= is the prefix for the VCC libraries.  Note that this
  is specified with no trailing period (e.g., SYS2.VCC rather
  than SYS2.VCC.).  Library names are assumed to be of the
  form vprefix.ASM, vprefix.MACLIB, etc.

VMICS= adds the MICS level to your data set names. If you do
  not want to include the additional level in your data set
  names ensure that VMICS is null (that is, VMICS=). If you
  want the MICS level added to your data set names ensure
  that VMICS is set to "MICS." (that is, VMICS='MICS.').

  NOTE:  Users must exit the data set yourprefix.MICS.CNTL
         for the VCCLOAD job to complete.


Submit the VCCLOAD job and verify the output of the job.

___  7. Now the data sets for VCC have been loaded from tape.
        From here on, the procedure is the same as in a full
        CA MICS site. Continue with the full checklist in the
        previous section 4.1.3.1.
```

# Chapter 5: OPERATION

VCC collects and records detailed DASD space utilization information from the following:

- Standard OS VTOCs
- Indexed VTOCs
- Standard VSAM master and user catalogs
- ICF BCSs and VVDSs

This chapter discusses the processing flow of VCC.

This section contains the following topics:

# 5.1 Operation Overview

VCC schedules a user-selectable number of subtasks that are responsible for reading VTOCs or VSAM/ICF catalogs in a way that will minimize contention with other subtasks and to maximize the distribution of VCC I/O across the DASD configuration.

VCC consists of a number of load modules. The basic functions are the following:

- Initialization, subtask scheduling, and termination reporting
- Parameter parsing
- VTOC scan subtask
- VSAM/ICF and VVDS catalog scan control subtask
- Standard VSAM/ICF BCS and VVDS catalog scan subtask
- VCC interface to DFHSM ARCUTIL data collection module
- VCC record writer

The VCC logic flow is shown in Figure 5-1.

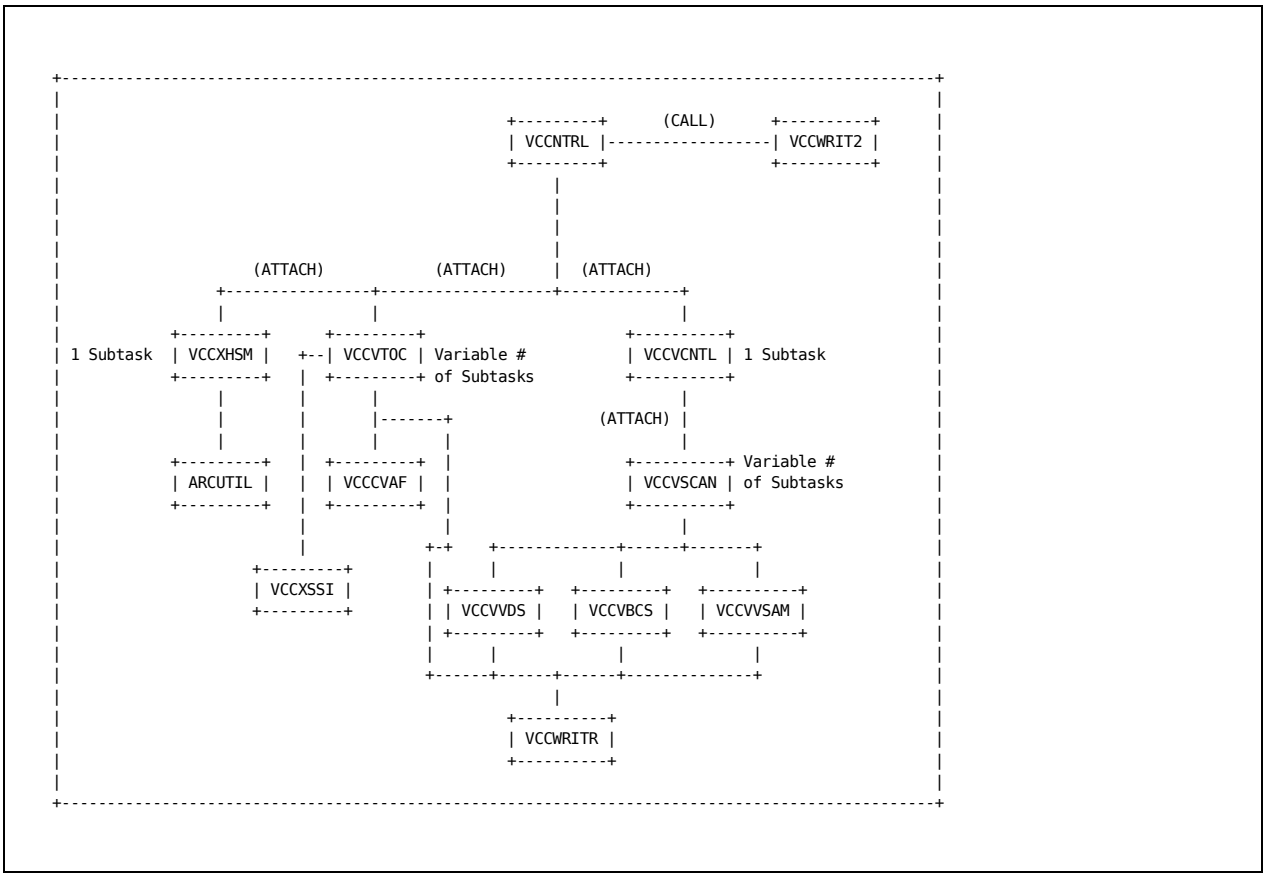



Figure 5-1. VCC Operation Overview

### 5.1.1 VCCNTRL - Initialization, Control and Reporting

The VCCNTRL module is responsible for VCC initialization, subtask ATTACH control, and VCC reporting after all processing has been completed. VCCNTRL performs the following functions:

- Acquires the VCC Common Work Area (CNTWA). This work area is addressable by all of the VCC modules and contains DCBs, parameter data set specifications, queue anchors, and general storage areas.

- OPENs the VCCVTMSG, VCCVSMSG, VCCSTATS, VCCDATA (if VCC recording to a data set has been requested), and VCCSNAP data sets. RDJFCBs are issued to make sure that the DD cards are present. In the case of the VCCDATA data set, the disposition is forced to be DISP=OLD via OPEN TYPE=J.

- LOADs all of the VCC modules that will be called as subroutines by the various VCC tasks.

- Calls VCCPARSE to read in and parse the parameter data set VCCPARMS (if specified).

- Scans the MVS DASD UCBs and builds a volume table that contains entries for every DASD volume on the system.

- Validates and OPENs the VCCNTROL data set. If this data set has not been initialized (is empty), then the previous VCC process timestamp is calculated to be 24 hours prior to the current VCC run timestamp. Otherwise, the VCCNTROL data set is read in, the previous VCC run timestamp is set to be equal to the last VCC run timestamp, the volume table is updated with the previous process timestamp of each volume (as indicated in the VCCNTROL data set), and the configuration SMF record is written. This SMF record is equivalent in content to the volume table.

- Builds a table called the VTOC Scan Control Table (VSCT). This table consists of up to 16 primary physical channel headers. Each PCHAN header points to a list of DASD string elements that in turn contains unit elements for each device to be scanned. The VSCT is created for two reasons:

  - To provide a means by which VTOC and catalog I/O generated by VCC can be spread across the DASD

configuration. To do this, VCC must first know what the configuration looks like.

- To provide areas in which to maintain statistics on each device scanned.

The INCLUDEVOL/EXCLUDEVOL, INCLUDEDEV/EXCLUDEDEV, and INCLUDEDV4/EXCLUDEDV4 parameters control the devices in this table. If you wish to include or exclude DASD devices by device address, use INCLUDEDEV/EXCLUDEDEV if your installation uses 3-digit UCB device addresses, and INCLUDEDV4/EXCLUDEDV4 if you use 4-digit UCB device addresses. Only one of the following parameters may be specified for a VCC scan: INCLUDEDEV, EXCLUDEDEV, INCLUDEDV4, or EXCLUDEDV4.

o ATTACHes the VCCVCNTL (VSAM catalog scan control) subtask. VCCVCNTL will in turn ATTACH the number of VSAM/ICF/VVDS catalog scan subtasks specified by the VSAMSUBT parameter.

o ATTACHes the number of VCCVTOC (VTOC scan) subtasks that were specified by the VTOCSUBT parameter.

o Calls VCCWRITR at entry point VCCWRIT2 to write records queued in the writer buffers to the proper data sets. After all records have been written and all subtasks have terminated, VCCWRIT2 returns to VCCNTRL.

o VCCNTRL updates the VCCNTROL data set with current VCC run information and writes the VCC Run Status Report.

## 5.1.2 VTOC Scan Overview

VTOC scan is comprised of a variable number of VCCVTOC subtasks. The number of subtasks is based on the number of DASD paths available for distributing the VCC I/O load. VCC will default to eight subtasks. You can directly specify this number to a maximum of 32 by using the VTOCSUBT parameter.

The following sections describe these functions:

1 - VCCVTOC--VTOC Scan Subtask
2 - VCCCVAF--Indexed VTOC Information Retrieval
3 - VCCXSSI--SMS Subsystem Interface

### 5.1.2.1 VCCVTOC--VTOC Scan Subtask

The VCCVTOC module scans the VTOCs of DASD devices and records the data they contain. VCCVTOC performs the following functions:

- It continually selects devices whose VTOCs are to be scanned from the VSCT. This selection process is continued until no more devices remain in the VSCT. It performs device selection as follows:
  - On initial entry, this module selects a device from the string that has the most devices to process on either the first VSCT physical channel that has no active VCCVTOC subtasks operating on it, or the VSCT physical channel that has the lowest ratio of active VCCVTOC subtasks to devices left to process.
  - It then selects a device from a string on the same physical channel that has the most devices left to process. If no more devices remain to be processed on a physical channel, it selects a device from the physical channel with the most devices left to process (if there is a string with no active VCCVTOC subtasks operating on it).
- Once it has selected a device, the VTOC is ENQ'd with the RESERVE macro. Parameters such as BYPASSRES, RESRETRY, and RESFAILOK are used to control the reserve process, and the action to be taken if the RESERVE fails. After reserve processing completes, the VTOC is allocated and opened for input.
- The Format 4 DSCB is then read separately from the other DSCBs in order to determine the bounds of the VTOC, the number of DSCBs per track, and the number of used DSCBs (to terminate the scan). If the VTOC is indexed, it calls the VCCCVAF module to extract the number of free DSCBs and to build the volume freespace extent record from the VTOC index, because this information is not present in the main VTOC when an index is used.
- It then scans the VTOC by reading in DSCBs one track at a time. Each DSCB is processed only once. If Format 2 or Format 3 DSCBs are encountered first, their information will be saved until the Format 1 DSCB is found. At this time, a complete SMF record

can be built and buffered by calling the VCCWRITR module.

- If a data set of the form SYS1.VVDS.Vvolser is found, it is added to the CNQ, chained from the volume table for the device on which the VVDS is located. The VCCVVDS module will process these entries later.

- After a VTOC scan is complete, the VTOC is CLOSEd, deallocated, and DEQ'd (which removes the reserve), if necessary.

### 5.1.2.2 VCCCVAF--Indexed VTOC Information Retrieval

The VCCVTOC module calls the VCCCVAF module and it retrieves information from the VTOC Index that it needs to properly set up the scan of the main VTOC. VCCCVAF uses the Common VTOC Access Facility (CVAF) to obtain the following information:

- The number of free DSCBs in the VTOC.

- Detailed free space extent information. It uses this information to build the volume free space segments for the VCC VTOC SMF record. VCCCVAF will return only 255 free space extents in one call (CVAF restriction), and will notify the caller via a return code if more free space extents exist. The caller will then continue calling VCCCVAF until all the extents are received.

### 5.1.2.3 VCCXSSI--SMS Subsystem Interface

The VCCVTOC module calls the VCCXSSI module which retrieves the storage group for a given volume from the SMS address space if it is active.

## 5.1.3 VSAM/ICF Catalog Scan Overview

VCC performs the catalog scan by using a variable number of subtasks. Catalog scan is composed of a single catalog scan control subtask (VCCVCNTL) and up to 32 (selectable by the VSAMSUBT parameter) catalog scan (VCCVSCAN) subtasks of VCCVCNTL. The VSAM/ICF catalog scan is comprised of the following modules:

1 - VCCVCNTL--ICF/VSAM Catalog Scan Control
2 - VCCVSCAN--ICF/VSAM Catalog Allocation and Routing
3 - VCCVVSAM--VSAM Catalog Scan
4 - VCCVBCS--ICF Catalog Scan
5 - VCCVVDS--ICF/VSAM Volume Data Set Scan

### 5.1.3.1 VCCVCNTL--ICF/VSAM Catalog Scan Control

The VCCVCNTL module schedules the VSAM/ICF catalogs that are to be scanned by the VCCVSCAN subtask. VCCVCNTL's logic is as follows:

- Checks to see if the master catalog name was provided in the parameters. If not, the CAXWA chain (represents open catalogs) is scanned to determine the name of the master catalog and the device on which it is located. The CAXWA chain is chained from the CVT; the master catalog is represented by the last CAXWA control block on the chain (forward pointer = 0).

- GETMAINs work areas for the VCCVSCAN subtasks called Task Activity Areas (TAAs). The VSAMSUBT value is used to control the number of GETMAIN'd TAAs as well as the number of VCCVSCAN subtasks to ATTACH.

- The VCCVSCAN subtasks are ATTACHed.

- The first subtask begins scanning the master catalog. As it encounters user catalogs or BCS catalog pointers, the VCCVSCAN subtasks adds them to a list of catalogs called the Catalog Name Queue (CNQ). CNQs are chained from the volume entry in the volume table that VCCNTRL built during initialization.

- After it has scanned all catalogs, VCCVCNTL will POST VCCNTRL to indicate that catalog scan processing has been completed.

### 5.1.3.2 VCCVSCAN--ICF/VSAM Catalog Allocation and Routing

The VCCVSCAN subtask allocates, opens and closes VSAM/ICF catalogs and VVDSs, maintains the CNQ, and routes control to VCCVVSAM, VCCVBCS, or VCCVVDS, depending on the catalog type. The following describes the logic of the VCCVSCAN subtask:

- Obtains a catalog entry from the CNQ. If one cannot be found, VCCVSCAN waits to be posted by a VCCVSCAN subtask that has encountered a catalog or VVDS pointer during its scan.
- Issues SVC 26 (LOCATE) to have the catalog opened as a catalog by Catalog Management.
- Turns on the STEPCAT bit in the PCCB for the open catalog in order to eliminate the need to have STEPCATs to process catalogs as VSAM data sets.
- Allocates and opens catalogs or VVDSs as VSAM data sets.
- Determines the catalog type. If it is a VSAM catalog, VCCVSCAN calls the VCCVVSAM module to process it. If it is an ICF BCS, VCCVSCAN calls the VCCVBCS module to process it. Otherwise, it calls the VCCVVDS module.
- Closes and deallocates the catalogs and VVDSs.
- Turns off the STEPCAT bit in the PCCB.

NOTE: If a catalog or VVDS pointer has been found by VCCVVSAM (or VCCVBCS) it will branch to a service routine located in VCCVSCAN. This routine will add the catalog or VVDS name to the CNQ for the volume in which it is located. If any other VCCVSCAN subtasks are waiting for work, they will be posted so they may process the new entry.

### 5.1.3.3 VCCVVSAM--VSAM Catalog Scan

The VCCVVSAM module scans old style (pre-ICF) VSAM master and user catalogs, formats the VCC SMF record, and calls the VCCWRITR module, which processes the records for the VCCDATA/SMF file. A more detailed description of the VCCVVSAM functions follows:

- Old style VSAM catalogs are split into two parts: a high-key range and a low-key range. The high-key range contains a true name record for each data set present in the catalog. The true name records are read sequentially to obtain the addresses of the active detail records, which reside in the low-key range area. It sorts the addresses into physical record sequence to minimize the DASD arm movements within the low-key range area. VCCVVSAM selects the following records from the low-key range for processing:

  C - catalog record type that describes VSAM data set clusters.

  D - catalog record type that describes the data component of VSAM data sets.

  I - catalog record type that describes the index component of a VSAM data set.

  G - catalog record type that describes alternate indexes.

  E - catalog extension records for the above record types.

  U - catalog record type that describes user catalogs. When it encounters a user catalog in the master catalog scan, VCCVVSAM determines if the user catalog is excluded or included through the use of the INCLUDECAT or EXCLUDECAT parameters. If it is excluded, the record is ignored. If it is included, then a test is made to make sure that the device on which it resides is known to VCC (see Chapter 3, VCC Usage, for further information on defining DASD devices to VCC). If these tests are passed, then VCCVVSAM will call a VCCVSCAN routine to add the user catalog entry to the Catalog Name Queue (CNQ) for subsequent processing.

- VCCVVSAM records all data contained in these catalog records. Note that this also includes the catalog self-defining records at the front of the catalog. It does not record catalog recovery area information.

### 5.1.3.4 VCCVBCS--ICF Catalog Scan

The VCCVBCS module reads and records information in ICF Basic Catalog Structures (BCSs). An ICF BCS is a VSAM key-sequenced data set (KSDS) that contains static catalog information (data that is not changed frequently). The ICF catalog record formats are different in structure from those encountered in VSAM catalogs, but the record types remain basically the same as those described in the previous section.

This module scans a BCS sequentially in much the same way that a VSAM catalog is scanned by VCCVVSAM. The data generated is identical in format. The difference lies in the fact that volume, extent, and Access Method Data Statistics Block (AMDSB) information is kept in a VSAM volume data set on the device that a VSAM data set resides on, and therefore won't be generated at the same time as the BCS information for those data sets (see the description of the VCCVVDS module).

VCCVBCS performs the same logical operations as the VCCVVSAM module.

#### 5.1.3.5 VCCVVDS--ICF/VSAM Volume Data Set Scan

The VCCVSCAN calls the VCCVVDS module to scan an ICF VSAM Volume Data Set (VVDS). The VVDS contains device level information about VSAM data sets that reside on the volume that contains the VVDS. The VVDS is pointed to by up to 36 BCSs that have VSAM data sets on this volume defined in them. The VCCVVDS module performs the following functions:

- o Sequentially reads the VVDS and generates VCC VSAM data set SMF records for every VSAM data set described by the VVDS. This information includes:
  - The VSAM data set name.
  - The name of the BCS in which the data set is defined.
  - The detailed extents of each data set that is on the device on which the VVDS resides. Note that a VSAM data set may spread across a number of volumes. This means that there is an SMF record generated for every volume on which the data set resides, as well as another record that is generated when the data set is encountered in a BCS. The CA MICS VCA component later combines this information into one observation for the data set.

### 5.1.4 Miscellaneous Service Routines

The following sections describe the VCC service routines:

1 - VCCPARSE--Parameter Parse Routine
2 - VCCWRITR--Record Writer Routine
3 - VCCQSCAN--Common Parameter Queue Scan Routine

### 5.1.4.1 VCCPARSE--Parameter Parse Routine

The VCCPARSE module handles the parsing of the VCC run parameters. During VCC initialization, the VCCNTRL main task loads, calls, and deletes this module. The main functions of VCCPARSE are the following:

- Verify the presence of a VCCPARMS DD JCL statement. If one is not present, assign defaults, and return control to VCCNTRL.
- Open the VCCPARMS data set.
- Parse each parameter in the VCCPARMS data set as it is read in. The valid parameter formats are the following:
  - KEYWORD=number
    character string
    data set name
  - KEYWORD=(list)

    where list can be a list of characters or data set names separated by commas. The list may span VCCPARMS records.
- Close the VCCPARMS data set and set defaults for any unspecified parameters after all VCCPARMS records have been read.
- Return control to VCCNTRL.

### 5.1.4.2 VCCWRITR--Record Writer Routine

The VCCWRITR module is the VCC record writer. While it is distributed as one physical module, it can be thought of as two logical modules: VCCWRITR and VCCWRIT2.

Any VCC subtask that desires to write a record to a file or to SMF calls VCCWRITR. The following parameters are passed to it:

- The address of the DCB, if not using SMF.
- The address of the record to be written.
- The address of the VCCNTRL common work area.
- The address of a processing options switch.

For data records, the standard SMF record header is initialized. If SMF recording was requested, it writes the record to SMF using the SMFWTM macro. If an error occurs, control returns to the caller with an appropriate return code set in register 15. For messages, it extracts and prefixes the current time to the front of the message.

For records that are to be written to a DCB, the critical information is saved in a writer buffer, and control returns back to the calling subtask. The VCCWRIT2 routine will process the record at a later time. If the record to be written exceeds 1019 characters, status information will be saved in the buffer and the subtask will wait for VCCWRIT2 to write the record. If no writer buffers are available, VCCWRITR waits for .2 seconds and then tries again. The format of the writer buffer is as follows:

For records greater than 1019 characters:

```
DS XL1 Processing switch:
          1...  ....  This is a message
          .1..  ....  This is a data record
          ..1.  ....  Record exceeds 1019 bytes
DS XL4 Address of the DCB
DS XL4 Address of the Record
DS XL3 (unused)
DS AL4 Address of the ECB to be POSTed
DS FL4 ECB to be POSTed after write
```

```
For records less than 1020 characters:

        DS XL1 Processing switch:
                  1...  ....  This is a message
                  .1..  ....  This is a data record
        DS XL4 Address of the DCB
        DS XL? Record to be written
```

The VCCNTRL main task calls VCCWRIT2 after all its other work is done. It scans the queue of writer buffers looking for active ones. When it finds an active buffer, the record is written to the correct DCB and the buffer is freed. For records longer than 1019 characters, the subtask requesting the write is also POSTed. If VCCWRIT2 can find no records to write, it will wait .5 seconds and then try again. After all records have been written and all subtasks have ended, VCCWRIT2 returns to VCCNTRL.

### 5.1.4.3 VCCQSCAN--Common Parameter Queue Scan Routine

VCCQSCAN is a common routine that scans lists of values that VCCPARSE creates when it encounters KEYWORD=(list) parameters. VCCQSCAN is passed:

- The address of the list (all lists are chained from the common work area).
- The address of the string that is to be matched against an element on the list.
- The address of a work area used if full string match (non-generic) is to be performed.

If the scan is unsuccessful, VCCQSCAN branches to the return address plus four. For unsuccessful list scans, VCCQSCAN branches to the return address.

## 5.1.5 VCC Interface to DFHSM

If you code HSM=Y in VCCPARMS, VCCNTRL will ATTACH the module VCCXHSM as a subtask to perform data collection about DFHSM-owned data sets and volumes while VCCVTOC and VCCVSCAN subtasks are asynchronously scanning VTOCs and catalogs.

VCCXHSM serves as a driver and control module for a module called ARCUTIL. IBM provides ARCUTIL in the DFHSM product starting with DFHSM Version 2.5 and is available at Version 2.4 with proper enabling PTFs from IBM. Chapter 17 of the IBM publication "DFHSM Version 2 Release 5.0 Installation and Customization Guide" (SH35-0084) thoroughly documents the ARCUTIL feature.

ARCUTIL provides a standard method of collecting data from the DFHSM Migration Control Data Set (MCDS) and the Backup Control Data Set. An application program provides a parameter list and an OPENed DCB for the output file and ARCUTIL reads the MCDS/BCDS and writes records to the DD specified in the parameter list. VCCXHSM is the driving application program VCC uses to communicate with ARCUTIL. DCOLLECT, available in DFP Version 3.2 and above, also uses ARCUTIL to gather DFHSM data.

When IBM introduced ARCUTIL, they also added warnings that application programs should not read the MCDS and/or BCDS directly due to the probability of change in format with future releases.

VCCXHSM uses ARCUTIL as described above and also intercepts records formatted by ARCUTIL before they are written. During the intercept, they are reformatted as follows for the stated reasons:

- Previous and current collection time stamps are added to the common segment of the record to allow CA MICS to compute DURATION, a critical element in an inventory-based measurement system.

- All variable length character strings are changed by stripping the field of blank padding characters and attaching the data elements as variable length strings. The elements subject to this compaction process are:

  - Data set name (DSNAME)
  - Management class (MGMTCLAS/STORMGTC)
  - Storage class (STORCLAS/STORSTGC)

- Data class (DATACLAS/STORDATC)

This compaction technique of VCCXHSM provides for substantial DASD space savings compared to the unmodified format of ARCUTIL-generated records (for example, those output by DCOLLECT).

## 5.1.6 VCC User Exits

At specific points within its processing flow, VCC allows for optional user written exit routines to receive control in order to add additional information, modify existing data in the output records, or in some way alter VCC's normal logic flow.

Samples of the user exits are distributed for your convenience. They illustrate coding techniques, register and linkage conventions, and entry and exit logic. These exits have been tested and work successfully. They are not however, certified in the same sense as the rest of VCC and you should carefully review the code in your own environment.

The following sections describe the various exit points provided by VCC:

1 - VTOC DSCB Exit (VCCUEXIT)
2 - DFHSM User Exit (VCCUHSMX)
3 - HFS User exit (VCCUSERX)

### 5.1.6.1 VTOC DSCB Exit (VCCUEXIT)

VCC allows you to extract site-dependent information from the Format 1 DSCB and to build a user segment in the VCC output record that is based on this information. Module VCCUEXIT is called after the record has been built and just before the VCCWRITR routine is called to write it to SMF or to the VCCDATA data set. VCCVTOC expects no return codes from the VCCUEXIT module.

An example of the VCCUEXIT is provided in the library sharedprefix.MICS.ASM as member VCCAEXIT. The source provided is in Assembler Language. You may copy VCCAEXIT, modify it to meet your specifications, assemble and link edit the object deck as VCCUEXIT.

NOTE: Saving the source code for your modified version of VCCUEXIT into sharedprefix.MICS.ASM as VCCAEXIT is not recommended as this member may be updated or replaced by future maintenance, thus removing your modifications. If you have made modifications to the source code, copy your updated member to sharedprefix.LOCALMOD.CNTL and document it.

In the sample code we distribute, VCCUEXIT retrieves a few bytes from the Format 1 DSCB used by Sterling Software's DMS/OS and arranges them in a user segment.

For more details regarding linkage conventions, work areas, and general purpose register usage, see the sample source member provided in sharedprefix.MICS.ASM(VCCAEXIT).

Review the macro definitions carefully before you start any coding. If you would like some work space, you can add WORK field definitions under the DSECT labeled SUSRDSCT. Simply define work fields after the EQUATE that establishes the length of the user segment.

Use fields SBASUSRS (user segment offset), SBASUSRL (user segment length), and SBASUSR# (number of user segments) as the reference points for the user segment. You must place the user segment at the end of the VCC record, and update the VCC record RDW (SBASRDW) with the user segment length that is being added to the value that is already in SBASRDW. Make sure that fields SBASUSRS, SBASUSRL, and SBASUSR# are set properly when adding a user segment as invalid results may cause VCC and/or VCA to fail. Do not update the VCCRCORD macro to reflect the user segment field definition layout; rather, create your own DSECT in the VCCUEXIT module. This

ensures that any maintenance to VCC will not conflict with the user segment definition. For more information regarding the VCC record layout, see section 5.2.

After user information has been picked up from the Format 1 DSCB (or elsewhere), you should return control to VCCVTOC using normal OS return logic as shown in the sample provided. Module VCCVTOC will then write the SMF record with the user segment added on to it.

Note the following:

- Do not modify existing fields within the VTCWA area mapped by the VCCVTCWA MACRO. This could cause a VCC failure.

- The Format 4 DSCB is kept in the VTCWA DSECT in field VTCWDSCB. It is 140 bytes in length.

- Certain fields in the base segment are not available to the VCCUEXIT module because they are filled in by the common VCCWRITR module. Therefore, do not attempt to use the following fields in the VCCUEXIT:

  SBASTYPE, SBASTIME, SBASDATE, SBASYSID, SBASLEN, SBASPSTR, SBASVID, SBASVERS.

- Specifying USEREXIT=Y at execution time in VCCPARMS for the Base VCC Collection process triggers the LOADing of a load-module called VCCUEXIT.

- You must code VCCUEXIT as reentrant and link-edit it as RENT,REUS because each concurrently executing VCCVTOC subtask calls this exit.

- Because VCC runs "authorized", this exit obtains information from protected areas by issuing the MODESET macro. If the protection key is set to zero, make sure that it is reset to nonzero (NZERO) before control is returned to VCCVTOC.

- The Sample source distributed has been tested and works successfully in a DMS/OS Release 1.8.1 environment using Sterling defaults. For other releases of DMS/OS, review the source code and appropriate Sterling documentation carefully.

### 5.1.6.2 DFHSM User Exit (VCCUHSMX)

The VCC DFHSM interface provides an exit point when module ARCUTIL passes information to VCC but before VCC writes its output record for DFHSM data. The user exit can add site dependent data to the output record and/or decide if the record should be written. Module VCCXHSM will expect a return code to be placed in register 15 by the user exit upon return.

An example of the VCCUHSMX is provided in the library sharedprefix.MICS.ASM as member VCCUHSMX. The source provided is in Assembler Language. You may copy VCCUHSMX, modify it to meet your specifications, assemble and link edit the object deck as VCCUHSMX.

NOTE: Saving the source code for your modified version of VCCUHSMX into sharedprefix.MICS.ASM as VCCUHSMX is not recommended as this member may be updated or replaced by future maintenance, thus removing your modifications. If you have made modifications to the source code, copy your updated member to sharedprefix.LOCALMOD.CNTL and document it.

In the sample code we distribute, VCCUHSMX notifies VCC not to write migrate records for data sets in ML2, and adds a user segment containing site dependent data to the output record.

For more details regarding linkage conventions, work areas, and general purpose register usage, see the sample source member provided in sharedprefix.MICS.ASM(VCCUHSMX).

Use fields SBASUSRS (user segment offset), SBASUSRL (user segment length), and SBASUSR# (number of user segments) as the reference points for the user segment. You must place the user segment at the end of the VCC record, and update the VCC record RDW (SBASRDW) with the user segment length that is being added to the value that is already in SBASRDW. Make sure that fields SBASUSRS, SBASUSRL, and SBASUSR# are set properly when adding a user segment as invalid results may cause VCC and/or VCA to fail. Do not update the VCCRCORD macro to reflect the user segment field definition layout; rather, create your own DSECT in the VCCUHSMX module. This ensures that any maintenance to VCC will not conflict with the user segment definition. For more information regarding the VCC record layout, see section 5.2.

Note the following:

- Certain fields in the base segment are not available to the user exit because they are filled in by the common VCCWRITR module. Therefore, do not attempt to modify the following fields:

  SBASTYPE, SBASTIME, SBASDATE, SBASYSID, SBASLEN, SBASPSTR, SBASVID, SBASVERS.

- The VCC VTOC Collector loads module VCCUHSMX and passes control to it for this exit point. If use of this exit point is not needed, make sure that there is no module within the load library with the name VCCUHSMX.

### 5.1.6.3 File Systems User exit (VCCUSERX)

The USS File Systems Collector provides an exit point when records are to be written to the VCCDATA file or to SMF. The user exit may add site dependent data to the output record and/or decide if the record should be written. VCC will expect a return code to be placed in register 15 by the user exit upon return.

An example of the VCCUSERX is provided in the library sharedprefix.MICS.ASM as member VCCUSERX. The source provided is in Assembler Language. You may copy VCCUSERX, modify it to meet your specifications, assemble and link edit the object deck as any name you choose. The name of the module is then supplied to VCC via the USEREXIT keyword.

NOTE: Saving the source code for your modified version of VCCUSERX into sharedprefix.MICS.ASM as VCCUSERX is not recommended as this member may be updated or replaced by future maintenance, thus removing your modifications. If you have made modifications to the source code, copy your updated member to sharedprefix.LOCALMOD.CNTL and document it.

For more details regarding linkage conventions, work areas, and general purpose register usage, see the sample source member provided in sharedprefix.MICS.ASM(VCCUSERX).

Use fields SBASOUSG (user segment offset), SBASOUSL (user segment length), as the reference points for the user segment. You must place the user segment at the end of the VCC record, and update the VCC record RDW (SBASRDW) with the user segment length that is being added to the value that is already in SBASRDW. Make sure that fields SBASOUSG and SBASOUSL are set properly when adding a user segment as invalid results may cause VCC and/or VCA to fail.

Note that certain fields in the base segment are not available to the user exit because they are filled in by VCC later in its processing flow. Therefore, do not attempt to modify the following fields:

SBASTYPE, SBASTIME, SBASDATE, SBASYSID, SBASLEN, SBASPSTR, SBASVID, SBASVERS.

# 5.2 VCC Data Recording

VCC records VTOC and catalog data to a data set. The VCCDATA DD statement references this data set in the JCL that is used to run VCC. It must be a disk data set. You can define it using sample JCL member VCCALLOC in library sharedprefix.MICS.CNTL.

The size of this data set is dependent on the number of DASD devices and number of data sets to be scanned by VCC. Generally, a z/OS system with about 900 DASD devices online will use approximately 45 cylinders of space on an IBM 3390.

Note that if you run VCC with SERIALIZE=Y you will cause VCC to issue an ENQ on SYSVTOC, which will raise a hardware RESERVE on the device to all connected systems. This will cause a problem with VCC itself if VCC is in the process of scanning the volume upon which the VCCDATA file is allocated and output processing requires expansion into another extent.

Therefore, use a larger primary allocation and no secondary allocation if you plan to operate with SERIALIZE=Y.

Detailed information on record layout can be found in the following subsections:

1 - VCC Data Record Layout
2 - VCC-DFHSM Data Record Layout

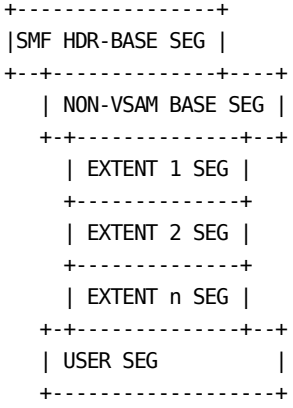
## 5.2.1 VCC Data Record Layout

The record that VCC generates is segmented in structure. In this way one record can represent several different types of data. The types include the following:

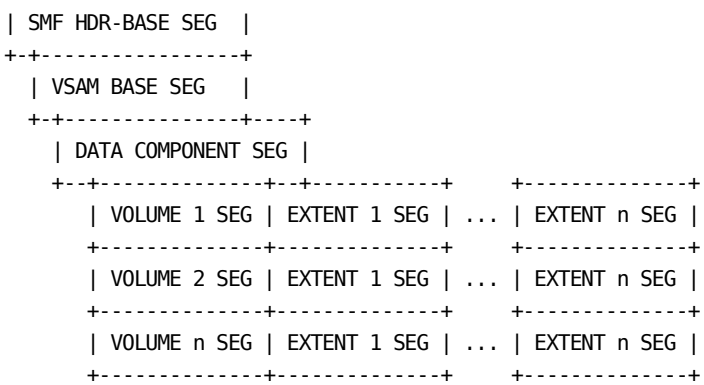
- The VTOC data set record
- The VSAM data set record
- The VTOC/volume record

All VCC-DFHSM records have a standard SMF header section. The VCCRCORD macro, distributed in sharedprefix.MICS.MACLIB, maps the records. The record layout is presented segment by segment as a box diagram. Detailed information at the field level is available by browsing the VCCRCORD macro in the latest release of the product.

VTOC Data Set Record




VSAM Data Set Record



```
    If needed:

    +--+--------------+---+
    | INDEX COMPONENT SEG |
    +--+--------------+---+----------+     +--------------+
       | VOLUME 1 SEG | EXTENT 1 SEG | ... | EXTENT n SEG |
       +--------------+--------------+     +--------------+
       | VOLUME 2 SEG | EXTENT 1 SEG | ... | EXTENT n SEG |
       +--------------+--------------+     +--------------+
       | VOLUME n SEG | EXTENT 1 SEG | ... | EXTENT n SEG |
       +--------------+--------------+     +--------------+
```

```
VTOC/Volume Record

+-----------------+
|SMF HDR-BASE SEG |
+-+---------------+-+
  | VTOC/VOLUME SEG |
  +----+------------+------------+
       | FREE SPACE EXTENT 1 SEG |
       +-------------------------+
       | FREE SPACE EXTENT 2 SEG |
       +-------------------------+
       | FREE SPACE EXTENT n SEG |
       +-------------------------+
```

## 5.2.2 VCC-DFHSM Data Record Layout

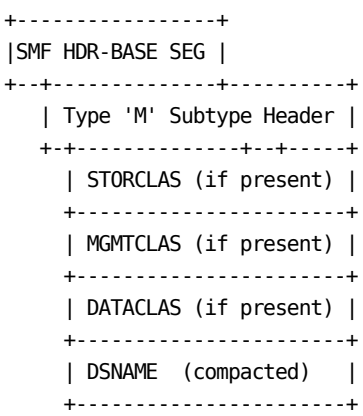
VCCXHSM generates a record that is variable length and designed for optimum data compaction consistent with some recognition requirements of the DFHSM portion of the CA MICS Space Analyzer Option. The record structure is much simpler than the format of the VCC records. The following record types are written:

- Migrated data set record (type 'M')
- Backup version record (type 'B')
- DASD inventory record (type 'C')
- Tape inventory record (type 'T')
- Inventory trailer record (type 'Z')

All VCC-DFHSM records have a standard SMF header section. The VCCRCORD macro, distributed in sharedprefix.MICS.MACLIB, maps the records. The record layout is presented segment by segment as a box diagram. You may browse the VCCRCORD macro in the latest release of the product to see detail information at the field level.

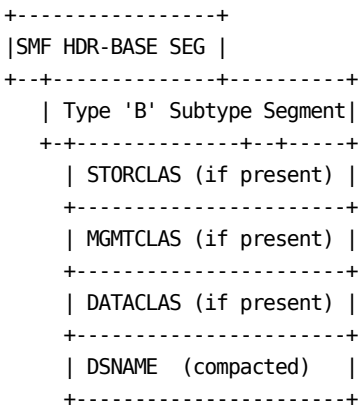
Migrated Data Set Record

The MCDS creates one type 'M' record for each migrated data set managed by DFHSM.




Backup Version Record

The BCDS creates one type 'B' record for each backup version of a data set managed by DFHSM.



```
DASD Inventory Record

One type 'C' record is created for each DASD volume for each
day of DFHSM activity retained in VSR records of DFHSM
control data sets.

+-----------------+
|SMF HDR-BASE SEG |
+--+--------------+----------+
   | Type 'C' Subtype Segment|
   +-+--------------+--+-----+

Tape Inventory Record

One type 'T' record is created for each "type" of tape
managed by DFHSM.  Currently there are 3 types - incremental
backup volumes, migrate volumes, and full volume dump
volumes.

+-----------------+
|SMF HDR-BASE SEG |
+--+--------------+----------+
   | Type 'T' Subtype Segment|
   +-+--------------+--+-----+
```

# 5.3 Abnormal Termination Processing

VCC is designed to terminate after most abends. Recovery routines are provided for all VCC tasks. The recovery routines provide a means by which VTOC/catalog scan subtasks can capture a system dump, close OPEN data sets, unallocate VTOCs and catalogs, and properly post VCCNTRL to avoid never-ending wait conditions.

In cases where the abend is determined to be non-recoverable, the abending subtask is terminated. Subtask termination generally results in job step termination unless you request that some number of critical errors can be ignored (see Section 5.3.1).

In those situations where the abend is determined to be recoverable, a subtask abend is treated as a non-critical error. For example, if a volume does not have a VTOC, an abend S213 will be reported in the VCCVTMSG data set and the subtask will attempt recovery from the abend. If recovery is successful, the subtask will clean up the processing environment, release held resources, generate error messages as necessary, and then continue processing with the next DASD volume to be scanned.

If an abnormal condition has occurred and the WTO(Y) parameter is in effect, VCC termination processing will issue WTO messages to the master console with a descriptor code of 2, informing the operator that a problem occurred and requesting a check of the VCC output for further information (see the documentation for message VCCCN13I in Chapter 7, Return Codes and Messages).

You should identify and correct problems that may have caused VCC to abend before it is run again. You can do this by excluding a problem volume or catalog from processing. In all cases, unless the problem is an operator CANCEL, an abend Sx37, or an abend S213, you should report the problem to CA Technical Support at http://ca.com/support.

The following sections describe special situations:

- Operator Cancel
- System Crash
- S0CX Abends
- SB37 and SE37 Abends
- The SNAP Output

### Operator Cancel

When a console operator cancels VCC, all processing is terminated. You may have to restart the VCC job at the backup step to save what data has already been collected and to turn off the "backup-required" flag in the control file. Rerun VCC to collect the remaining data before the CA MICS VCA DAILY processing starts.

### System Crash

A system crash is hard to recognize and is beyond the scope of VCC. Therefore, after the system comes back up and before the CA MICS VCA DAILY processing starts, you may have to restart the VCC job at the backup step to save the data that has already been collected and to reset the "backup-required" flag in the control file. You should then run VCC again to collect the remaining data before the CA MICS VCA DAILY processing starts. It is the responsibility of the operations staff or the programmer who is running VCC to make sure that this is done.

### S0Cx Abends

VCC itself most likely causes these types of abends. In these cases, the SDUMP macro is invoked to schedule a system dump capture to a SYS1.DUMPxx data set, and the abending subtask is terminated. The person responsible for VCC should contact CA Technical Support for problem assistance and a possible fix for the problem.

### Sx37 Abends

These abends are caused by the inability to obtain additional DASD extents for the VCCDATA data set. When this condition occurs, all processing is terminated. You may have to restart the VCC job at the backup step to save the data that has already been collected and to turn off the "backup-required" flag in the control file. You should re-allocate the VCC VCCDATA data set and rerun the VCC job to collect the remaining data before the CA MICS VCA DAILY processing starts.

### SNAP Output

VCC may issue a SNAP macro to the VCCSNAP data set if it detects an invalid output record.  This is rare. If this condition occurs, the output should be saved and sent to CA Technical Support for analysis.

Note that VCC will no longer write snap dumps to the VCCSNAP data set for the affected address space during recovery processing.

### 5.3.1 Ignoring Critical Errors With VSAMERR or VTOCERR

A critical error is any non-recoverable abend that occurs during the processing of a VTOC or catalog. After VCC successfully OPENs either a VTOC or a catalog data set and writes the first record to SMF or to VCCDATA for that volume or catalog, an "in-progress" switch is set by VCC. Later, during recovery processing of a subtask abend, if the "in-progress" flag is on, the error is considered critical. This causes recovery processing to set the "critical-error" flag just prior to completing subtask termination.

The "critical-error" flag is significant because it is designed to halt the entire VCC processing run. As the remaining active subtasks finish scanning a VTOC or catalog, they look for the next unit of work to perform. Before starting a new unit of work, each subtask checks to see if a critical error has occurred. If the "critical-error" flag has been set, each subtask terminates and the main task will terminate with a return code of 16.

The VSAMERR=n and VTOCERR=n keyword parameters allow you to specify that "n" number of critical error conditions should be permitted before forcing termination of the entire VCC processing run. After "n" critical error conditions have been permitted, recovery processing for a failing task will set the "critical-error" flag that signals all remaining active subtasks to terminate at the completion of the current unit of work.

For example, assume you coded VSAMERR=1 and VSAMSUBT=4. This causes four subtasks to be created to scan the BCS and VVDS structures in the system. VCC relies on the structural integrity of the cells in the BCS and VVDS and, if an erroneous cell exists, it is highly likely that some VCC module will abend with an S0C4. Since VSAMERR=1 is coded, it will ignore this critical error. The subtask terminates and the remaining three subtasks continue to look for work to perform and complete normal processing.

The catalog or VTOC being scanned by the failing subtask is marked in error, and other tasks of the same type will not attempt to rescan the object.

Continuing the example, if another VSAMSUBT (subtask) abends, the VSAMERR limit of 1 will have been exceeded. In this case, the "critical-error" flag will be set to signal all remaining active subtasks to terminate at the first opportunity.

The principle behind this function is that it is better to allow the bulk of the data collection to proceed to normal completion with the understanding that some small percentage of the data is incomplete and will be gathered later after appropriate diagnostic work has been done on the catalog or VTOC that caused the problem.

It is important to realize that the subtask in error abends and reduces the number of remaining tasks. Therefore, the VSAMSUBT value must always be at least 1 greater than the value specified for VSAMERR (and the same for VTOCSUBT and VTOCERR).

## 5.3.2 Sending SYS1.DUMP Data Sets to CA

VCC issues the SDUMP macro in ESTAE processing to capture the system and address space status for diagnosis with the IPCS facility.

Unformatted dumps are captured in SYS1.DUMPxx data sets along with a descriptive message to the console and JES log of the job. A descriptive title is associated with the dump if it was issued by a VCC ESTAE routine.

Follow these procedures to help CA diagnose the problem:

1. Save the dump from being cleared by operations.

   There are a limited number of SYS1.DUMPxx data sets in any installation. When all SYS1.DUMPxx data sets are FULL, operators are prompted and existing dumps are often cleared to make way for the new dumps.

   Use CA PREVAIL/COMMAND product with the DUMPDS command to find the dump(s) or you can do the following:

   a. Issue PDF 3.4 for SYS1.DUMP* to find the suffixes used in your installation.

   b. Locate the dump written to by VCC. The title should contain CA MICS SPACE COLLECTOR. Another way of finding exactly which SYS1.DUMPxx data set to save is to match the title information seen in the JES log for the failing VCC job with the outputs of the following TSO command:

      ```
      SYSDSCAN aa:bb
      ```

      where:
      aa and bb are a range of suffixes. Example:

      ```
      SYSDSCAN 80:88
      ```

      This example would show the dump information for SYS1.DUMP80 through SYS1.DUMP88.

   c. Having located the dump, copy it to a safe location with IEBGENER. Example:

      ```
      //COPYDUMP EXEC PGM=IEBGENER
      //SYSPRINT DD SYSOUT=*
      //SYSUT1   DD DISP=SHR,DSN=SYS1.DUMP18
      ```

```
//SYSIN    DD DUMMY
//SYSUT2   DD DSN=userid.SAVEDUMP.MICS.VCC,
//            SPACE=(CYL,(9,9),RLSE),
//            DISP=(NEW,CATLG,DELETE)
```

d. Having copied the dump, the SYS1.DUMPxx data set can be cleared at any time.

2. Report the problem to CA Technical Support.

a. There may be sufficient information in the JES Log and other VCC outputs (e.g. VCCVTMSG, VCCVSMSG and so on) to resolve the problem.

There may be a known symptom for which a fix or bypass is available or there may be certain installation-specific factors that may help resolve the problem.

b. If CA Technical Support needs the dump in order to diagnosis the problem, copy the dump to a tape cartridge and mail it to CA. The support representative can fax you exact mailing directions.

Continuing with the example above:

```
//FILE1    EXEC PGM=IEBGENER
//SYSUT1   DD DSN=userid.SAVEDUMP.MICS.VCC,DISP=SHR
//SYSUT2   DD DISP=(,PASS),UNIT=3480,
//            LABEL=(1,SL,EXPDT=98000),
//            DSN=VCC.DUMP
//SYSPRINT DD SYSOUT=*
//SYSIN    DD DUMMY
```

Notes:

- the tape does NOT have to be standard label.
- the DSNAME does not have to be as shown.
- please choose a cartridge we will NOT have to return to you.

# Chapter 6: MAINTENANCE

This chapter discusses the policies and procedures for maintaining the VCC product.

This section contains the following topics:



## 6.1 Maintenance Policies

The following maintenance policies are recommended for timely support of VCC.

- You should apply distributed maintenance within 120 days of receipt.
- We distribute product changes on magnetic tapes. Each product change provides a complete description of the problem it is fixing, explains how the problem is fixed, gives instructions on how to apply the fix, and lists the fix in IEBUPDTE format.
- Occasionally, large development efforts will result in a new release. This new release will be distributed as a COMPLETE REPLACEMENT for the existing VCC product version. It will have incorporated into it all maintenance that has been made available through the date of the new release.
- Integration of maintenance and user modifications is your responsibility.
- Do not resequence any provided source code, macro, or JCL members. All maintenance is distributed for these members according to the statement sequence numbers maintained by the CA MICS Product Support Group.

# 6.2 Applying Maintenance

This section describes how to apply maintenance to VCC. It covers the following topics:

1 - Applying Maintenance to the Source Code
2 - Assembling and Link Editing VCC Maintenance

## 6.2.1 Applying Maintenance to the Source Code

You should integrate VCC maintenance by using IBM's IEBUPDTE utility. The following example shows how to apply maintenance to the VCC source code using IEBUPDTE:

```
//jobname  JOB  ...
//STEP1    EXEC PGM=IEBUPDTE
//SYSPRINT DD   SYSOUT=A
//SYSUT1   DD   DISP=OLD,DSN=sharedprefix.MICS.ASM
//SYSUT2   DD   DISP=OLD,DSN=sharedprefix.MICS.ASM
//SYSIN    DD   DSN=sharedprefix.MICS.LOCALMOD.CNTL(CTF),
//               DISP=SHR
```

## 6.2.2 Assembling and Link Editing VCC Maintenance

CA Technical Support no longer supplies updates to VCC source code requiring on-site assembly. New load modules or control statements for the IBM Utility AMASPZAP are provided.

Sample JCL member VCCCOPY in library sharedprefix.MICS.CNTL is used to copy new versions of load modules provided by a product change. The following example shows how to apply maintenance to VCC load modules using AMASPZAP:

```
//jobname  JOB  ...
//STEP1    EXEC PGM=AMASPZAP,PARM=IGNIDRFULL
//SYSPRINT DD SYSOUT=A
//SYSLIB   DD DISP=SHR,DSN=sharedprefix.MICS.LOAD
//SYSIN    DD DISP=SHR,DSN=sharedprefix.MICS.CNTL(VCCnnnnA)
```

# Chapter 7: RETURN CODES AND MESSAGES

VCC produces return codes or error messages for every error it encounters. It writes messages to the SYSPRINT, VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG DDnames in the following format:

```
VCCnnnns   message_text
or
VCCmmnns   message_text
```

where:

mm is a two-character value defining the VCC module that issued the error message. These values are:

- BC - VCCVBCS
- BL - VCCBLDVT
- CN - VCCNTRL
- HS - VCCXHSM
- PR - VCCPARSE
- SC - VCCVSCAN
- SM - VCCXSMS
- VC - VCCVCNTL
- VD - VCCVVDS
- VS - VCCVVSAM
- VT - VCCVTOC
- XD - VCCDIAG

nn or nnnn is the identifying number of the message.

s is the severity level of the message. The values are:

- I - Informational message
- W - Warning message
- E - Critical error message
- A - Console operator action/response message

Definitions of the error messages are in Section 7.2. They are organized in alphabetical order (ignoring the severity level character of the message identifier).

More detailed information is available in the following subsections:

1. Return Codes Description
2. Messages Description

# 7.1 Return Codes Description

VCC produces return or condition codes upon execution completion as an aid in checking the success of its processing. The possible return codes are listed below by the primary modules that provide the return codes to the Operating System:

VCCNTRL and VCCMAIN Return codes

0 Successful completion with no errors encountered.

4 One or more warning messages were issued. This return code is intended to alert you to the need to analyze the VCC output Message log(s) for possible errors that may affect the results of the CA MICS Space Analyzer run (VCA). Because of the various ways in which control keywords may modify the processing of VCC, such as VTOCERR, VSAMERR, EXCLUDEVOL, etc, you may need to examine the output Message log(s) and VCC statistics reports to determine if this is a complete and successful run. Because CA MICS (VCA) will reject data set level records that do not have associated volume level records, this data collection run can be ignored.

8 An error has occurred. In most cases the data collection is incomplete. Analyze the VCC output Message log(s) to determine the cause of the error, repair it and rerun VCC or contact CA Technical Support.

12 A severe environmental error has occurred. Check the VCC output message log(s) and the Job log to determine the cause of the error. Ensure that all required JCL statements are properly specified. After the cause of the error is corrected, rerun VCC. If the problem persists contact CA Technical Support.

16 A severe operational error has occurred. Check the Job Log and other output to determine the cause of the error. Ensure that all required JCL statements are properly specified. This return code is usually issued when VCC was not installed properly. Review both jobs and installation procedures used to install VCC. After correcting the problem, rerun

VCC. If the problem persists contact CA Technical Support.

20 The required VCCSTATS/SYSPRINT DD statements are not found within the JCL, or an abnormal condition (ABEND) was detected. If you are unable to correct the problem contact CA Technical Support.

VCCUPDT Return Codes

0 Successful completion with no errors encountered.

8 VCCNTROL DD statement is missing.

12 Open failed for the VCCNTROL data set.

16 GETMAIN failed for the VCCUPDT work area.

VCCUTIL Return Codes

0 Successful completion with no errors encountered.

4 One or more warning messages were issued.

8 One or more error messages were issued. Processing is terminated. Analyze the VCC Utility output Message log (SYSPRINT) to determine the cause of the error, repair it and rerun VCC or contact CA Technical Support.

12 A critical error has occurred during processing.

16 An operational error has occurred. Ensure that all required JCL statements are specified. This return code is usually issued when VCC was not installed properly. Review both jobs and installation procedures used to install VCC. After correcting the problem rerun VCC. If the problem persists contact CA Technical Support.

20 The required SYSPRINT DD statement was not found within the JCL or an abnormal condition (ABEND) was detected.

# 7.2 Messages Description

This section describes the messages VCC produces and the situations that can cause VCC to issue each message. The following sections describe these messages:

1 - VCCVBCS Messages
2 - VCCNTRL Messages
3 - VCCPARSE Messages
4 - VCCVSCAN Messages
5 - VCCVCNTL Messages
6 - VCCVVSAM Messages
7 - VCCVTOC Messages
8 - VCCXHSM Messages
9 - VCCMAIN Messages
10 - VCCDIAG Messages

## 7.2.1 VCCVBCS Messages

```
VCCBC01I
--------

TASK(nn)  VOL=Volser rrrrrrrrrrrr.  VOLUME SEGMENT
          INCOMPLETE FOR DSN=data set name.

NOTE:  This message will only appear if the TEST=Y parameter
has been specified. Its main function is VCC error diagnosis.
VCC issues this message when the BCS scan module VCCVBCS
determines that a multi-volume VSAM data set is defined on a
volume that is excluded from VCC processing because of one of
the following reasons (rrrrrrrrrrrr):

   EXCLUDED      - Volume has been excluded from VCC
                   processing directly using the EXCLUDEVOL
                   parameter, or indirectly by not being in a
                   VCC INCLUDEVOL list (if specified).

   NOT MOUNTED   - Volume is not mounted and online, and
                   therefore is not being processed by VCC.

   NOT READY     - Volume is mounted, but was not ready at
                   the time the VCC internal volume table was
                   built, and therefore is not being
                   processed by VCC.

   MOUNTED WRONG - Volume is mounted, but does not have the
                   desired mount attributes to be eligible to
                   be processed by VCC (refer to Section
                   3.1.1.2 for descriptions of REMOVEMNT and
                   RESERVEMNT).

VCCVBCS action:  VCCVBCS does not rebuild the volume
                 segment for the volume in question.
```

```
VCCBC02W
--------

TASK(nn)  VOL=Volser rrrrrrrrrrrr.   NO PROCESSING FOR USER
          CATALOG=user catalog name.

NOTE:  This message will only appear if the TEST=Y parameter
has been specified, as its main function is VCC error
diagnosis.  VCC issues this message when the BCS scan module
VCCVBCS finds a user catalog connector that points to a
volume that is excluded from VCC processing because of one of
the reasons listed for message VCCBC01I.

VCCVBCS action:  VCC does not process the user catalog.


VCCBC03W
--------

TASK(nn)  BUFFER OVERRUN.  INCOMPLETE RECORD WRITTEN FOR
          DSN=data set name.

VCC issues this message if the log record currently being
created is larger than 4089 bytes (SMF recording) or the
record length of the VCCDATA data set (data set recording).

VCCVBCS action:  The record is written.  Any data that would
                 have made the VCC log record longer than
                 the maximum (including the RDW) is lost.

VCCBC04I
--------

TASK(NN) NO PROCESSING FOR EXCLUDED VSAM DSN=data set name.

NOTE:  This message will only appear if the TEST=Y parameter
has been specified, as its main function is VCC error
diagnosis.  VCC issues this message if the current catalog
record being processed defines a VSAM data set that has been
explicitly excluded through the EXCLUDEDSN parameter or
implicitly excluded by not being in the INCLUDEDSN list (if
any).

VCCVBCS action:  The catalog record is ignored.
```

```
VCCBC05E
--------

TASK(nn)  SMF WRITE ERROR.  CATALOG PROCESSING TERMINATING.

VCC issues this message if an error occurs while a VCC log
record is being written to SMF.

VCCVBCS action:  VCC terminates processing.

VCCBC14W
--------

TASK(nn)  BCS CATALOG READ FAILURE.  R15.ccccccc
          REASON.rrrrrrrr

VCC issues this message if an error occurs while VCCVBCS is
attempting to read a BCS catalog record.  The contents of R15
are displayed (ccccccc) as well as the error reason code
(rrrrrrrr).  Refer to IBM publication VSAM Administration:
Macro Instruction Reference manual for a description of error
return codes and reason codes.

VCCVBCS action:  VCC processing is terminated.
```

## 7.2.2 VCCNTRL Messages

```
VCCBL01I
--------

Diagnostic dump for Unit(XXXX) Volume(VVVVVV)

VCC issues this message when an error has been detected while
trying to access information from the control unit for the
indicated volume (vvvvvv), and the TRACE=Y parameter was
specified.  'xxxx' indicates the device address.  The
information provided assists CA Technical Support personnel
in determining the cause of the problem.  Refer also to
VCCBL03E.
```

```
VCCBL03W
--------

I/O error on Unit(XXXX) Volume(VVVVVV) IOSCOD(AA)
CSW(BBBBBBBB ZZZZZZZZ)

VCC issues this message after several unsuccessful attempts
to communicate with the DASD storage subsystem or control
unit that volume 'vvvvvv' is attached to.  'xxxx' indicates
the device address.  'zzzzzz' is diagnostic information
returned in the Channel Status Word (CSW) to aid CA Technical
Support personnel in determining the cause of the problem. CA
Technical Support personnel may direct you to rerun VCC with
the TRACE=Y run-time parameter to provide further diagnostic
information. You can also check with your local Technical
Support personnel to determine if the DASD storage subsystem
has the most recent microcode level.

VCCNTRL action:  Processing continues. Message VCCCN26W will
                 indicate the action taken by VCC for the
                 volume shown within the message.

                 Note: Some information regarding the device
                 that contained the error may not be
                 available in the VCCDATA record. This can
                 affect other CA MICS components that use
                 this data, such as the Accounting and
                 Chargeback Option and the Space Analyzer
                 Option (VCA).
```

```
VCCCN06E
--------

Load failed for Module(XXXXXXXX)

VCC issues this message if the LOAD for a VCC module fails,
and that module is required for proper VCC operation.  A LOAD
failure can occur for one or more of the following reasons:

   o  The modules are missing from the VCC STEPLIB data
      set.

   o  The required STEPLIB DD statement is missing from the
      VCC JCL.

   o  The modules are missing from the linklist.

VCCNTRL action:  Processing is terminated until the situation
                 that caused the error is corrected.


VCCCN07E
--------

Volume Table getmain failed; Increase region size and rerun

VCC issues this message if the GETMAIN for the DASD
configuration list and the VSCT fails (return code from
GETMAIN ^= 0).  The GETMAIN is done from subpool zero;
therefore, to correct this situation you need to increase the
VCC JCL region size.

VCCNTRL action:  Processing is terminated until the situation
                 that caused the error is corrected.
```

```
VCCCN09E
--------

Getmain failed for control file buffer; increase region size
and rerun

VCC issues this message if the GETMAIN for the control file
I/O buffer fails.  This message is applicable only when SMF
is not the VCC recording medium (SMF=N).  The size of the
buffer is the larger of:

    o  The largest SMF physical record (4089 bytes for VCC).

    o  The control file logical record length.

Increase the region size on the EXEC statement and rerun
VCC.

VCCNTRL action:  Processing is terminated until the situation
                 that caused the error is corrected.


VCCCN10E
--------

Backup of VCCDATA data set has not been performed.
Perform backup and rerun

VCC issues this message only if SMF is not the VCC recording
medium (SMF=N).  In this situation, the third step of the VCC
job did not complete successfully.  Make sure that the backup
of the VCCDATA data set is done (this creates the SMF file
that goes into the CA MICS DAILY run) and that the third step
of the VCC job is run (this clears the "do backup" flag in
the VCCNTROL data set).

VCCNTRL action:  Processing is terminated until the situation
                 that caused the error is corrected.
```

```
VCCCN11I
--------

Volume(VVVVVV) was processed during a previous VCC run but is
not in the current configuration

The VCCNTRL module has encountered a volume in the control
file that is not online in the current DASD configuration.
This message is informational only.  You should research
this situation to make sure that DASD devices are not
mistakenly being missed by VCC.

VCCNTRL action:  Processing continues normally.


VCCCN12I
--------

The following set of error messages is displayed under the
common identifier of VCCCN12I at the master console through
the use of the Write To Operator (WTO) facility. The
specification of the WTO run-time parameter enables this
action.  The purpose of this VCC feature is to notify the
operator that VCC encountered some abnormal conditions that
may need to be examined further.  VCC issues the following
WTO messages if problems (or potential problems) occur during
VCC processing:

   ZZ VTOC scanning subtasks abended. Keep the
           SYS1.DUMP__ data sets for diagnosis.

   ZZ VSAM scanning subtasks abended. Keep the
           SYS1.DUMP__ data sets for diagnosis.

VCCNTRL action:  VCC termination processing continues.

See Section 5.3.2 for specifics on handling SYS1.DUMP
data sets.

The following WTO message is issued if VCC runs without
encountering any unusual problems:

   VCC PROCESSING TERMINATED NORMALLY.

VCCNTRL action:  VCC termination processing continues.
```

VCCCN13I
--------

The following set of error messages is displayed under the common identifier of VCCCN13I at the master console through the use of the Write To Operator (WTO) facility. The specification of the WTO run-time parameter enables this action. The purpose of this VCC feature is to notify the operator that VCC encountered some abnormal conditions that may need to be examined further. VCC issues the following WTO messages if problems (or potential problems) occur during VCC processing:

One or more WARNING messages have been issued

One or more ERROR messages have been issued

VCCNTRL action: VCC termination processing continues.

See Section 5.3.2 for specifics on handling SYS1.DUMP data sets.

The following WTO message is issued if VCC runs without encountering any unusual problems:

VCC PROCESSING TERMINATED NORMALLY.

VCCNTRL action: VCC termination processing continues.

VCCCN14W
--------

VCCVTOC work area getmain failed; Increase region size and rerun

VCC issues this message if the GETMAIN in a VCCVTOC subtask fails. Because other VCCVTOC subtasks may already have obtained their work areas, this is a warning message. If no data was collected, increase the region parameter on the EXEC statement and rerun VCC.

VCCNTRL action: VCC termination processing continues.

```
VCCCN15E
--------

Internal error; Insufficient space for tables related to
devices; Issuing SDUMP

Save the SYS1.DUMPxx created by this message. Then increase
REGION size and rerun.


Note:
-----
VCC generates messages VCCCN16E through VCCCN22E when the
VCCNTRL module determines that the VTOC/catalog scan subtasks
had a problem writing records to the z/OS SMF data set.
These problems will not occur if VCC is using a separate data
set as the VCC recording medium.  These messages are as
follows:


VCCCN16E
--------

VCC SMF record RDW is less than 18 bytes

VCC issues this message if an SMF record created by VCC is
less than 18 bytes in length (including the RDW).  This is a
VCC problem and should be reported to CA Technical Support.


VCCCN17E
--------

SMF is not active

This message is issued if SMF recording is not active. Report
this message to your system programmer.
```

```
VCCCN18E
--------

IEFU83 SMF exit rejecting VCC SMF records

VCC issues this message if the IEFU83 SMF exit is rejecting
SMF records with the VCC SMFID (default is 198).  If this is
occurring, make certain that the exit is modified to accept
VCC SMF records.


VCCCN19E
--------

VCC SMF data record has been lost

VCC issues this message if all the SYS1.MAN files are full
and no more SMF records can be written.  Report this problem
to your system programmer.


VCCCN20E
--------

VCC SMF record number is not being recorded; Check SMFPRMXX
parmlib member

VCC issues this message if the SMFPRMXX SYS1.PARMLIB member
used in the current IPL specifies that SMF records with the
VCC SMFID are to be ignored.  Modify SMFPRMXX to enable VCC
SMF recording.


VCCCN21E
--------

SMFWTM SVC could not establish error recovery

VCC issues this message if the SMF writer SVC (SVC 83) could
not establish recovery.  Inform your system programmer.

VCCNTRL action:  VCC termination processing continues.
```

```
VCCCN22E
--------

VCC SMF record not written due to SMF buffer shortage

VCC issues this message if SMF does not have enough buffers
to handle the record output rate currently being experienced
by the system.  Inform your system programmer.

VCCNTRL action:  VCC termination processing continues.


VCCCN23E
--------

VCC could not write a record to SMF

VCC issues this message if the GETMAIN for VCCWRITR buffers
fail.  Because the size of this area is determined by the
value of the NUMWTRBUF parameter and the number of subtasks
specified, you may want to either increase the region size or
adjust those parameters.

VCCNTRL action:  VCC termination processing continues.


VCCCN24E
--------

VCCNTRL module is not APF authorized; Re-link module with
AC=1 into an APF authorized load library

VCC issues this message if the VCCNTRL module is not APF-
authorized.  VCC must run as an authorized task to perform
many of its tasks.  This error can be caused by not placing
VCC modules in an authorized library, or by not coding the
AC=1 option when using the linkage editor to link module
VCCNTRL.

VCCNTRL action:  VCC termination processing continues.
```

```
VCCCN25W
--------

SMS managed volumes were found online with SMS inactive;
Storage group set to blanks

VCC issues this warning message at normal termination if the
SMS address space was unavailable for the extraction of
storage group information.  The storage group is set to
blanks in the record.  If SMS is inactive, but no SMS-managed
volumes were detected during the scan, this message is not
produced.


VCCCN26W
--------

Volume(VVVVVV) on Device(DDDD) has an unknown device type;
MMMMMMMMMMMMMM

VCC was unable to obtain device hardware information for
volume (vvvvvv).  'hhhh' is the device unit address. If the
message indicates that the volume was excluded, this can
indicate that the hardware device is unsupported by VCC, or
the DASD storage subsystem does not have the most recent
microcode level.  If the microcode level is current and the
problem persists after rerunning VCC, contact CA Technical
Support for further direction.

VCCNTRL action:  The message indicates the action taken for
                 the volume shown. If the message indicates
                 that defaults are used, this can affect the
                 results of other CA MICS components such as
                 the Accounting and Chargeback Option and the
                 Space Analyzer Option (VCA).
```

```
VCCCN27I
--------

Stalled subtasks detected; Proceeding with main task
termination

An abnormal condition has occurred. During termination of the
main task of VCC, a last minute check of the TCB chain has
revealed that one or more subtasks are still present.  This
is indicative of some sort of error in either the VSAM or
VTOC subtasks and probably traceable to messages written by
either VCCVSCAN or VCCVTOC. Rather than WAIT any further for
subtask termination that may never occur, VCC produces and
writes this message to the console log.  It will get the
operator's attention since it requires manual deletion (it
will not roll off automatically.)

VCCNTRL action:  VCC will issue the message to the console
                 only (not VCCSTATS) and then proceed to
                 terminate normally.  z/OS will abend the
                 job with an S-A03 abend code.

The true error can be traced back to a subtask that has begun
but for some reason has not finished.  Examine the VCCVTMSG
and VCCVSMSG logs and the VCCSTATS report to identify the
object (VTOC or catalog) where VCC processing failed to
finish.


VCCCN28W
--------

Volume(VVVVVV) not found in the VCCNTROL file; Default
duration will be assigned to all data sets on this volume

vvvvvv is either a brand-new volume serial number to this VCC
configuration or vvvvvv is back online after having been
offline in the prior VCC run.  This is a warning message to
alert you that the VCA default DURATION will be used for all
data sets on this volume.

VCCNTRL action:  VCC will issue the message and set the
                 previous VCC run time for this volume to
                 zero, which triggers the default duration
                 processing in VCA.
```

```
VCCCN29E
--------

SRM DONTSWAP failed for collector address space

Some IBM operating services that are invoked by the VCC
Space Collector Option require the VCC address space to be
temporarily non-swappable.  A request to the SYSEVENT
DONTSWAP service to request non-swappable status failed.

VCCNTRL action:  VCC terminates with a condition code of 12
                 since it is unable to utilize certain system
                 services.


VCCCN32I
--------

Data collection "backup-required" flag set in control file

The "backup-required" flag has been set and hardened to disk
in the VCC control file.  The collection file backup and
control file reset steps must be completed successfully prior
to the next VCC Space Collector Option execution.

VCCNTRL action:  VCC continues with termination processing.
```

```
VCCCN33E
--------

CA MICS Space Collector ABEND zzzz Rsn(rrrrrrrr)
Mod(mmmmmmmm) Info(iiiiiiii)

This message is issued if an abend occurs in the main task
of VCC.  The message appears in the VCCSTATS log if it is
OPEN and is always written to the JES user log and the
console.

VCCNTRL action:  A complete SVC dump is written to a
                 SYS1.DUMPxx data set.  If VCCSTATS is OPEN,
                 a complete status report is written to
                 VCCSTATS.  Scanning is probably incomplete.
                 Volumes not scanned and catalog structures
                 not scanned are indicated by the message
                 flag **NOT READ** in the rightmost column of
                 the report.

Report the problem to CA Technical Support. Save the
SYS1.DUMPxx data set and the complete VCC job including
VCCSTATS, VCCVTMSG, VCCVSMSG, and VCCHSMSG.


VCCCN34W   or   VCCVT34W
--------        --------

Failure to unpin device number hhhh; Vol(vvvvvv) Ret(dd)
Rsn(ee)

See the description for this message in section 7.2.7
(VCCVT34W).


VCCCN35I
--------

Normal termination resource recovery complete

Processing for the VCC Space Collector Option has completed
normally, and recovery and cleanup of all critical resources
have been completed successfully.
```

```
VCCCN36I
--------

Abnormal termination resource recovery complete

Processing for the VCC Space Collector Option has completed
following an abnormal condition.  Recovery and cleanup of
all critical resources have been completed successfully.
```

### 7.2.3 VCCPARSE Messages

```
VCCPR01E
--------

GETMAIN FOR VCCPARSE WORK AREA FAILED.

The GETMAIN for the VCCPARSE work area fails.  Increase the
region size and rerun VCC.

VCCPARSE action:  Control is returned to VCCNTRL with
                  register 15 set to a nonzero value (8).


VCCPR02I
--------

VCCPARMS DD CARD MISSING.  DEFAULT VALUES WILL BE USED.

The RDJFCB for the VCCPARMS DD statement indicates that the
DD statement is missing.  This message is informational.

VCCPARSE action:  All parameters that have defaults have
                  these default values set in the CNTWA.
                  Control is then returned to VCCNTRL.
```

```
VCCPR03E
--------

OPEN ERROR ON VCCPARMS DATA SET.

The VCCPARMS DCB indicates that the OPEN for the VCCPARMS
data set failed.  This error is critical because it is
assumed that since a VCCPARMS DD card is present, the
installation intended to exercise some control over the way
VCC collects DASD space data.

VCCPARSE action:  Control is returned to VCCNTRL with
                  register 15 set to a nonzero value (8).

VCCPR04E
--------

UNKNOWN KEYWORD.

A parameter keyword was encountered but VCCPARSE could not
find it in the parameter table.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.


VCCPR05E
--------

MISSING EQUALS SIGN.

A parameter keyword has been identified but it is not
immediately preceded by an equals sign (with no intervening
blanks).

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.
```

```
VCCPR06E
--------

VALID OPTIONS FOR KEYWORD ARE Y AND N.

A parameter keyword is encountered whose only valid options
are Y (yes) and N (no), and one of these values was not
entered.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.

VCCPR07E
--------

KEYWORD OPTION MUST BE NUMERIC.

A parameter keyword whose only valid options are numeric
values is contains an option that is not numeric.

VCCPARSE action:  Parameter keyword parsing continues. When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.


VCCPR08E
--------

KEYWORD OPTION VALUE TOO LARGE.

A parameter keyword whose only valid options are numeric
values is encountered, but the numeric value entered is too
large for VCCPARSE to handle (more than ten digits and/or
over 2147483647).

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.
```

```
VCCPR09E
--------

KEYWORD OPTION VALUE OUT OF ALLOWABLE RANGE.

A parameter keyword is encountered whose only valid options
are numeric values and the numeric value entered is outside
of the range of values allowed for this keyword.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.

VCCPR10E
--------

KEYWORD OPTION VALUE NOT ALLOWED.

A parameter keyword is encountered that has a list of valid
numeric or character values associated with it, but the value
entered is not in this list.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.


VCCPR11E
--------

KEYWORD OPTION VALUE TOO LONG.

A parameter keyword is encountered whose options have a
maximum length associated with them and one of the options
entered is over this maximum length.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.
```

```
VCCPR12E
--------

INVALID DATA SET NAME.

A parameter keyword is encountered whose options must be data
set names and one of the options entered is not a valid data
set name.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.

VCCPR14E
--------

MISSING LEFT PARENTHESIS.

A parameter keyword is encountered whose options must be
enclosed in parentheses and the left parenthesis is missing.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.


VCCPR15E
--------

MISSING RIGHT PARENTHESIS.

A parameter keyword is encountered whose options must be
enclosed in parentheses and the right parenthesis is missing.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.
```

```
VCCPR16E
--------

INVALID OPTION LIST SEPARATOR.

A parameter keyword whose options must be enclosed in
parentheses contains a character assumed to be a separator
character and this character is not a comma.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.

VCCPR17E
--------

GETMAIN FOR KEYWORD LIST AREA FAILED.

A parameter keyword is encountered whose options must be
enclosed in parentheses and the GETMAIN for the area that is
to contain this specified list of values fails.  Increase
region size and rerun VCC.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.



VCCPR18E
--------

KEYWORD OPTION TERMINATOR MUST BE BLANK.

A parameter keyword is encountered that can have only one
option and the option specified does not end in a blank
character.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.
```

```
VCCPR19E
--------

MISSING KEYWORD OPTION.

A parameter keyword is encountered that has no value
specified after the equals sign and no intervening blanks.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.

VCCPR20E
--------

INVALID OPTION LIST SEPARATOR CHARACTER

A parameter keyword is encountered whose option must be a
data set name or other character value and that parameter
keyword is not terminated by a comma (,), period (.), or
right parenthesis ()).

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.


VCCPR21E
--------

MISSING OPTION TERMINATOR.

A parameter keyword is encountered whose option does not
terminate on the same card.  For keywords with a list of
options, one of the items in the list does not terminate on
the same card on which the options began.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.
```

```
VCCPR22E
--------

KEYWORD MUTUALLY EXCLUSIVE WITH ANOTHER.

A parameter keyword is encountered whose mutually exclusive
counterpart has been encountered previously.  This applies to
all INCLUDE/EXCLUDE parameter keywords.

VCCPARSE action: Parameter keyword parsing continues.  When
                 parameter parsing completes, control is
                 returned to VCCNTRL with register 15 set to
                 8.

VCCPR23E
--------

GENERIC SPECIFICATION NOT ALLOWED FOR THIS KEYWORD.

A parameter keyword is encountered whose operand contains an
asterisk.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.


VCCPR24E
--------

NUMBER OF TOTAL ENTRIES FOR KEYWORD EXCEEDS nnn.

A parameter keyword is encountered whose options are a list
of values, and the number of entries specified in this list
exceeds the maximum allowed for the keyword. The string nnn
in the message text is replaced by the maximum number of
entries defined for the parameter keyword currently being
parsed.

VCCPARSE action:  Parameter keyword parsing continues.  When
                  parameter parsing completes, control is
                  returned to VCCNTRL with register 15 set to
                  8.
```

```
VCCPR25W
--------

PREVIOUS OPTION IS OBSOLETE AND SHOULD BE REMOVED FROM
PARAMETER INPUT.

A parameter keyword is encountered that has been obsoleted
and VCC no longer uses.

VCCPARSE action:  The parameter keyword and its value are
                  ignored and parameter keyword parsing
                  continues.

VCCPR26W
--------

CATCHECK=Y WAS REQUESTED.  SERIALIZE=N IS BEING FORCED TO
PREVENT EXCESSIVE RESERVE TIMES.

VCC performance is slowed by the CATCHECK function since
LOCATE processing is done to the catalog for each VTOC data
set to be written out to VCCDATA or SMF.  Since the scan of
any VTOC will therefore be slower with the function, it is
important to avoid serializing on the SYSVTOC resource while
issuing LOCATE SVCs.  This combination of SERIALIZE=Y and
CATCHECK=Y is not allowed.

VCCPARSE action:  SERIALIZE=N is forced into effect to allow
                  CATCHECK processing to proceed.


VCCPR27W
--------

VTOCERR VALUE HAS BEEN RESET TO 1 LESS THAN THE NUMBER OF
VTOC SUBTASKS.

The count of critical VTOC scanning errors that can be
ignored cannot be greater than the number of VTOC scanning
tasks (VTOCSUBT).

VCCPARSE action:  The error threshold is reset to 1 less
                  than the number of subtasks which leaves
                  one subtask remaining if the error
                  threshold is reached.
```

```
VCCPR28W
--------

VSAMERR VALUE HAS BEEN RESET TO 1 LESS THAN THE NUMBER OF
VSAM SUBTASKS.

The count of critical VSAM scanning errors that can be
ignored cannot be greater than the number of VSAM scanning
tasks (VSAMSUBT).

VCCPARSE action:  The error threshold is reset to 1 less
                  than the number of subtasks which leaves
                  one subtask remaining if the error
                  threshold is reached.

VCCPR29E
--------

SMFID VALUE AND HSMSMFID VALUE CANNOT BE THE SAME NUMBER WHEN
LOGGING TO SMF.

Both VCC records and DFHSM records are to be logged to SMF
but the SMF record numbers cannot be identical.  Either the
defaults should be used or logging should be directed to one
or both of the corresponding data sets (VCCDATA or VCCHSM).

VCCPARSE action:  The run is terminated immediately so this
                  error can be resolved and the job submitted
                  again.
```

## 7.2.4 VCCVSCAN Messages

```
VCCSC01I
--------
                                ICF
TASK(nn) PROCESSING STARTED FOR VSAM CATALOG=dsn
                                VVDS

The VCCVSCAN module selects a CNQ to process.  The catalog
type and name are displayed.

VCCVSCAN action:  Catalog scan processing continues normally.



VCCSC02I
--------
                                  ICF
TASK(nn) PROCESSING SUCCESSFUL FOR VSAM CATALOG=dsn
                                  VVDS

The VCCVSCAN module receives control back from one of the
scan modules with a return code of zero (0).

VCCVSCAN action:  Catalog scan processing continues normally.



VCCSC04W
--------
                                      ICF
TASK(nn) PROCESSING NOT SUCCESSFUL FOR VSAM
                                      VVDS
         CATALOG=dsn

One of the scan modules has detected and printed an error
message.  The catalog processing is not successful.

VCCVSCAN action:  Catalog scan processing continues with the
                  next catalog.

VCCSC05W
--------

TASK(nn) CATALOG LOCATE FAILURE R15.cccccc REASON.rrrrrr
         DSN=dsn
```

```
While processing catalog DSN= one of the scan modules
received a bad return code from the LOCATE function.  Refer
to IBM publication VSAM Administration:  Macro Instruction
Reference for a description of error reason and return codes.

VCCVSCAN action:  Catalog scan processing continues with the
                  next catalog.



VCCSC06W
--------

TASK(nn) CATALOG ALLOCATION FAILURE R15.cccccc REASON.rrrrrrr
         DSN=dsn

While allocating catalog DSN=, one of the scan modules
received a bad return code from dynamic allocation.  Refer to
IBM publication System Macros and Facilities, Volume, 1 for a
description of dynamic allocation error reason and return
codes.

VCCVSCAN action:  Catalog scan processing continues with the
                  next catalog.



VCCSC07W
--------

TASK(nn) CATALOG MODCB FAILURE R15.ccccccc REASON.rrrrrrr
         DSN=dsn

While doing a MODCB for catalog DSN= one of the scan modules
received a bad return code.  Refer to IBM publication VSAM
Administration: Macro Instruction Reference for a description
of error reason and return codes.

VCCVSCAN action:  Catalog scan processing continues with the
                  next catalog.

VCCSC08W
--------

TASK(nn) CATALOG OPEN FAILURE R15.ccccccc REASON.rrrrrrr
         DSN=dsn

While opening catalog DSN= one of the scan modules received a
bad return code.  Refer to IBM publication VSAM
```

```
Administration:  Macro Instruction Reference for a
description of error reason and return codes.

VCCVSCAN action:  Catalog scan processing continues with the
                  next catalog.



VCCSC10W
--------

TASK(nn) DUPLICATE DEFINITION. VOL=vvvvvv,
         CATALOG=dsn

Note:  This message will only appear if the TEST=Y parameter
has been specified, as its main function is VCC error
diagnosis.  This message is issued when a catalog is defined
in more than one catalog.  The name of the catalog and the
volume on which it resides are displayed.

VCCVSCAN action:  Catalog scan processing continues normally.



VCCSC11I
--------

TASK(nn) NO PROCESSING FOR EXCLUDED USER
         CATALOG=dsn

A user catalog connector record is encountered during the
scan of a catalog and either the catalog name defined in the
catalog connector record or the volume on which the user
catalog resides has been excluded.

VCCVSCAN action:  Catalog scan processing continues normally.

VCCSC13E
--------

TASK(nn) CRITICAL ERROR DETECTED.  PROCESSING TERMINATING.

A VCCSCAN subtask finishes processing a catalog and
determines that a critical error has occurred in another
subtask.

VCCVSCAN action:  Catalog scan processing terminates.
```

```
VCCSC14E
--------

TASK(tt) ABEND COMPLETION CODE = S-hhh;
  SUBTASK TERMINATING SCAN OF dsname

  where:    tt      =  task number that abended
            S-hhh   =  System Abend code in hex
            Udddd   =  User Abend code in decimal
            dsname  =  SYS1.VVDS.V______ or other DSNAME


VCCVSCAN action:  Checking performed on VSAMERR values.
                  DUMP taken to a SYS1.DUMPxx data set (*).
                  Message echoed to console and JES log.
                  VSAM subtask terminates.

Note that a dump is only taken for the first VSAM subtask
that abends. If more than one subtask of this type abends,
the message will be written to document what structure was
being processed but the dump will be bypassed.

See section 5.3.2 for instructions on what to do with the
dumps if you receive this message.
```

## 7.2.5 VCCVCNTL Messages

```
VCCVC01I
--------

VSAM DATA COLLECTOR HAS COMPLETED PROCESSING.

This message is issued when VCCVCNTL has determined that no
more catalogs remain to be processed.  The message is issued
just before the ECB that indicates the completion of catalog
processing is POSTed.

VCCVCNTL action:  Catalog scan processing is terminated by a
                  POST to wake up the VCCNTRL main task.


VCCVC02I
--------

TASK(nn)  HAS COMPLETED SUCCESSFULLY/UNSUCCESSFULLY.
          TIME=HH:MM:SS.TH
     CATALOGS PROCESSED.###   WAITS ISSUED.www

This message is issued when a VCCVSCAN subtask has determined
that no more catalogs remain to be scanned, and terminates.
If no errors were encountered during the life of the
task, this message indicates that processing completed
successfully; otherwise, an unsuccessful completion will be
indicated.

VCCVSCAN action:  Catalog scan processing continues until all
                  other VCCVSCAN subtasks POST complete to
                  VCCVSCAN.



VCCVC03W
--------

VSAM COLLECTOR HAS TERMINATED WITH ERRORS.

This message is issued when a VCCVSCAN subtask abends while
processing a catalog, a BCS, or a VVDS.

VCCVSCAN action:  Catalog scan processing continues until all
                  other VCCVSCAN subtasks POST complete to
                  VCCVSCAN.
```

```
VCCVC04E
--------

TASK(nn) HAS COMPLETED ABNORMALLY.  CODE=Sccc
         TIME=HH:MM:SS.TH

VCCVCNTL issues this message when it finds that a VCCVSCAN
subtask has abended.  Sccc is the abend code.  A dump will
accompany this message.

VCCVSCAN action:  Catalog scan processing terminates.



VCCVC05W
--------

THE MASTER CATALOG RESIDES ON VOLUME vvvvvv WHICH IS EXCLUDED
FROM PROCESSING.

VCCVCNTL issues this message when it attempts to process the
Master Catalog but finds that the volume it resides on is
either not mounted or not ready.

VCCVSCAN action:  Catalog scan processing terminates without
                  any catalogs being scanned.



VCCVC06W
--------

VSAM GENCB FAILURE.  R15.ccccccc REASON.rrrrrrr

VCCVCNTL issues this message when a VCCVSCAN subtask
terminates and the return codes in the work area show that a
VSAM GENCB has failed.  R15.ccccccc is the value that
register 15 had when the error condition was encountered.
REASON.rrrrrrr is the value that register zero (0) had when
the GENCB failed.  For the meanings of the different reason
codes for GENCB failure, see IBM publication GC26-3838,
Access Method Services: VSAM Programmers Guide.

VCCVSCAN action:  Catalog scan processing continues with the
                  next catalog CNQ found.

VCCVC07W
--------
```

```
UNABLE TO ESTABLISH ESTAE ENVIRONMENT.  R15.ccccccc

VCCVCNTL issues this message if an ESTAE request failed.  The
reason code returned in register 15 is displayed.

VCCVSCAN action:  Catalog scan processing terminates without
                  any catalogs being scanned.



VCCVC08W
--------

WORK AREA GETMAIN FAILED.  R15.ccccccc.

VCCVCNTL issues this message if a GETMAIN request for a
VCCVSCAN work area fails.  It displays the reason code
returned in register 15.

VCCVSCAN action:  Catalog scan processing terminates without
                  any catalogs being scanned.



VCCVC09W
--------

VCC VOLUME TABLE ERROR.

VCCVCNTL issues this message when it validates the volume
table built by VCCNTRL.  This error message is issued if the
length is not positive.

VCCVSCAN action:  Catalog scan processing terminates without
                  any catalogs being scanned.
```

## 7.2.6 VCCVVSAM Messages

```
VCCVS01I
--------

TASK(NN) VOL=Volser rrrrrrrrrrrr. VOLUME SEGMENT INCOMPLETE
          FOR DSN=data set name.

NOTE:  This message will appear only if the TEST=Y parameter
have been specified, as its main function is VCC error
diagnosis.  This message is issued when the VSAM catalog scan
module VCCVVSAM determines that a multi-volume VSAM data set
is defined on a volume that is excluded from VCC processing
because of one of the following reasons (rrrrrrrrrrrr):

   EXCLUDED      - Volume has been excluded from VCC
                   processing directly using the EXCLUDEVOL
                   parameter, or indirectly by not being in a
                   VCC INCLUDEVOL list (if specified).

   NOT MOUNTED   - Volume is not mounted and online, and
                   therefore is not being processed by VCC.

   NOT READY     - Volume is mounted, but was not ready at
                   the time the VCC internal volume table was
                   built, and therefore is not being
                   processed by VCC.

   MOUNTED WRONG - Volume is mounted, but does not have the
                   desired mount attributes to be eligible to
                   be processed by VCC (refer to Section
                   3.1.1.2 for descriptions of REMOVEMNT and
                   RESERVEMNT).

VCCVVSAM action:   VCCVVSAM does not build the volume segment
                   in question.



VCCVS02W
--------

TASK(NN) VOL=volser rrrrrrrrrrrr.  NO PROCESSING FOR USER
          CATALOG=user catalog name.

NOTE:  This message will only appear if the TEST=Y parameter
has been specified, as its main function is VCC error
```

diagnosis. This message is issued when the VSAM catalog scan module VCCVVSAM finds a user catalog connector that points to a volume that is excluded from VCC processing because of one of the reasons listed for message VCCVS01I.

VCCVVSAM action: VCC does not process the user catalog.

VCCVS03W
--------

TASK(NN) BUFFER OVERRUN.
DSN=data set name.

VCC issues this message if the log record currently being created is larger than 4089 bytes.

VCCVVSAM action: The record is written. Any data that would have made the VCC log record larger than 4089 bytes in length (including the RDW) is lost.

VCCVS04I
--------

TASK(NN) NO PROCESSING FOR EXCLUDED VSAM DSN=data set name.

NOTE: This message will appear only if the TEST=Y parameter has been specified, as its main function is VCC error diagnosis. This message is issued if the current catalog record being processed defines a VSAM data set that has been explicitly excluded through the EXCLUDEDSN parameter or implicitly excluded by not being in the INCLUDEDSN list (if any).

VCCVVSAM action: The catalog record is ignored.

VCCVS05E
--------

TASK(NN) SMF WRITE ERROR. CATALOG PROCESSING TERMINATED.

This message is issued if an error occurs while a VCC log record is being written to SMF.

```
VCCVVSAM action:  VCC terminates processing.

VCCVS06W
--------

TASK(NN)  CLUSTER SETS-OF-FIELDS MISSING FOR DSN=data set
          name.

This message is issued if there are no sets-of-fields
pointers in the cluster record defining data set name.

VCCVVSAM action:  VCC writes an incomplete record.



VCCVS07W
--------

TASK(NN) VOLUME INFORMATION MISSING FOR USER CATALOG = user
          catalog name.

This message is issued if a catalog connector record is
encountered during the scan of a catalog, but the volume
information is missing.

VCCVVSAM action:  VCC cannot process the user catalog.



VCCVS08W
--------

TASK(NN) EXTENSION SETS-OF-FIELDS MISSING.
          DSN=data set name

This message is issued if the number of sets-of-fields in a
catalog extension record is zero.

VCCVVSAM action:  VCC writes an incomplete record for the
                  VSAM data set in question.


VCCVS09W
--------

          DATA
TASK(NN)  INDEX SETS-OF-FIELDS MISSING.  DSN=data set name.
```

This message is issued if the number of sets-of-fields in a catalog record defining the data or index component of a VSAM data set is zero.

VCCVVSAM action: VCC writes an incomplete record for the VSAM data set in question.

VCCVS10W
---

```
          DATA
TASK(NN)  INDEX COMPONENT POINT FAILURE.  R15.ccccccc
          REASON.rrrrrrrr
```

VCC issues this message if the POINT fails for the catalog records that describe the data/index component of a VSAM data set. The register 15 return code and the VSAM error reason code are provided. Refer to manual VSAM Administration: Macro Instruction Reference for a description of error reason and return codes.

VCCVVSAM action: VCC writes an incomplete record for the VSAM data set in question.

VCCVS11W
---

```
          DATA
TASK(NN)  INDEX COMPONENT READ  FAILURE.  R15.ccccccc
          REASON.rrrrrrrr
```

VCC issues this message if the READ fails for the catalog records that describe the data/index component of a VSAM data set. The register 15 return code and the VSAM error reason code are provided. Refer to manual VSAM Administration: Macro Instruction Reference for a description of error reason and return codes.

VCCVVSAM action: VCC writes an incomplete record for the VSAM data set in question.

VCCVS12W
---

```
TASK(NN)  EXTENSION RECORD POINT FAILURE.  R15.ccccccc
```

```
          REASON.rrrrrrrr

VCC issues this message if the POINT fails for a catalog
extension record.  The register 15 return code and the VSAM
error reason code are provided.  Refer to manual VSAM
Administration:  Macro Instruction Reference for a
description of error reason and return codes.

VCCVVSAM action:  VCC writes an incomplete record for the
                  VSAM data set in question.

VCCVS13W
--------

TASK(NN)  EXTENSION RECORD READ FAILURE.  R15.ccccccc
          REASON.rrrrrrrr

VCC issues this message if the READ fails for a catalog
extension record.  The register 15 return code and the VSAM
error reason code are provided.  Refer to manual VSAM
Administration:  Macro Instruction Reference for a
description of error reason and return codes.

VCCVVSAM action:  VCC writes an incomplete record for the
                  VSAM data set in question.



VCCVS14W
--------

TASK(NN)  VSAM CATALOG READ FAILURE.  R15.ccccccc
          REASON.rrrrrrrr

VCC issues this message if the READ for the next sequential
catalog record fails.  The register 15 return code and the
VSAM error reason code are provided.  Refer to manual VSAM
Administration:  Macro Instruction Reference for a
description of error reason and return codes.

VCCVVSAM action:  VCC writes an incomplete record for the
                  VSAM data set in question.



VCCVS15W
--------

TASK(NN)  VSAM CATALOG POINT FAILURE.  R15.ccccccc
```

```
          REASON.rrrrrrrr

VCC issues this message if the POINT for the next sequential
catalog record fails.  The register 15 return code and the
VSAM error reason code are provided.  Refer to manual VSAM
Administration:  Macro Instruction Reference for a
description of error reason and return codes.

VCCVVSAM action:  VCC writes an incomplete record for the
                  VSAM data set in question.
```

## 7.2.7 VCCVTOC Messages

```
VCCVT01E
--------

TASK(nn) Getmain failed for DSCB resolution queues; Subtask
         terminated

The GETMAIN for the storage area that holds the queues used
to resolve DSCB CCHHR chain pointers fails.  If you want to
run with the number of VTOC scan subtasks that were
specified, increase the region size and rerun VCC.

VCCVTOC action:  The subtask issuing the error message
                 terminates.  All other subtask processing
                 continues.


VCCVT02E
--------

TASK(nn) Unable to establish recovery environment; Subtask
         terminated

The ESTAEX issued by VCCVTOC fails. If a subtask runs without
abend recovery, unending waits can result; therefore, the
subtask terminates.

VCCVTOC action:  The subtask issuing the error message
                 terminates.  All other subtask processing
                 continues.
```

```
VCCVT03E
--------

TASK(nn) Getmain failed for I/O area; Subtask terminated

The GETMAIN for the area that builds VSAM data event control
blocks fails and VTOC scan processing cannot continue.
Increase VCC region size and rerun.

VCCVTOC action:  The subtask issuing the error message
                 terminates.  All other subtask processing
                 continues.

VCCVT04E
--------

TASK(nn) Getmain failed for I/O buffer; Subtask terminated

The GETMAIN for the area that VSAM uses as an I/O buffer to
read in a track's worth of DSCBs fails.  Increase the region
size and rerun.

VCCVTOC action:  The subtask issuing the error message
                 terminates.  All other subtask processing
                 continues.


VCCVT05W
--------

TASK(nn) Allocation error     - Vol(vvvvvv) Ret(eeeeeeee)
         Rsn(rrrrrrrr)

The allocation for the volume on which a VTOC resides fails.
For the error reason, information reason, and return code
meanings, see IBM publication GC28-0627, OS/VS2 MVS System
Programming Library: Job Management.

VCCVTOC action:  The subtask issuing the error message
                 terminates processing for the VTOC on that
                 volume and continues normally.
```

```
VCCVT06W
--------

TASK(nn) CVAF map volume error, Vol(vvvvvv) Ret(eeeeeeee)
         Rsn(rrrrrrrr)

A call to CVAF for volume free space extents fails.  You can
determine the volume serial number of the device by examining
the VCC VTOC scan log (the output on VCCVTMSG) and finding
the VCCVT10I message issued by the same task (nn).  VCCVTOC
is in the middle of building the VTOC/VOLUME SMF record free
space extents.

VCCVTOC action:  The subtask issuing the warning message
                 continues processing the VTOC as if it did
                 not have an index.  The SMF record is
                 written as is, with an error indication
                 turned on indicating that an error occurred.

VCCVT07W
--------

TASK(nn) CVAF map VTOC error  - Vol(vvvvvv) Ret(eeeeeeee)
         Rsn(ssssssss)

A call to CVAF for the number of free DSCBs in the VTOC
fails.  You can determine the volume serial number in the
same way as with message VCCVT06W.

VCCVTOC action:  The subtask issuing the warning message will
                 read the entire VTOC since the number of
                 used DSCBs cannot be determined.


VCCVT08W
--------

TASK(nn) VTOC not initialized - Vol(vvvvvv) volume not
         processed

The DASD volume specified by Vol(vvvvvv) did not contain a
valid Volume Table of Contents (VTOC); therefore, the scan
task is unable to process the volume.

VCCVTOC action:  The subtask issuing the error message will
                 terminate processing of the current VTOC and
                 continue with the next VTOC.
```

```
VCCVT09I
--------

TASK(nn) VTOC collection scan processing termination complete

A VCCVTOC subtask cannot find another VTOC to process; it
terminates.

VCCVTOC action:  The subtask issuing this message continues
                 with termination.

VCCVT10I
--------

TASK(nn) VTOC scan initiation - Vol(vvvvvv) Unit(uuuu)
         Device(dddddd)

A VCCVTOC subtask is about to begin processing a VTOC on
volume serial number vvvvvv.

VCCVTOC action:  The subtask issuing this message continues
                 processing the VTOC.


VCCVT11I
--------

TASK(nn) VTOC scan completion - Vol(vvvvvv)

A VCCVTOC subtask has completed processing a VTOC on volume
serial number vvvvvv.

VCCVTOC action:  The subtask issuing this message continues
                 processing the next VTOC (if any).
```

```
VCCVT12I
--------

TASK(nn) Reserve not obtained - Vol(vvvvvv) will be re-tried
         again later

A VCCVTOC subtask has tried three times to ENQ/RESERVE a VTOC
and has failed on each attempt.

VCCVTOC action:  The subtask will place this volume back on
                 the queue for later processing, and will
                 increment the RESRETRY value for this
                 volume.

VCCVT13E
--------

TASK(nn) Critical scan error; processing termination
initiated

A VCCVTOC subtask is about to select another VTOC for
processing but on examining the global error flags, finds
that a critical error has occurred, causing VCC to terminate
further processing.

VCCVTOC action:  The subtask issuing this message terminates.
```

```
VCCVT14E
--------

TASK(nn) Scan task ABEND zzzzz- Vol(vvvvvv) Rsn(rrrrrrrr)
         Data Collection Not Initiated

  where:    nn      =  Task number that abended
            zzzzz   =  System abend code in hex (Shhh) or
                       user abend code in decimal (Udddd)
            vvvvvv  =  Volume being addressed at time of
                       abend

VCCVTOC action:   Checking performed on VTOCERR values.
                  DUMP taken to a SYS1.DUMPxx data set (*).
                  Message echoed to console and JES log.
                  VSAM subtask terminates.

Note that a dump is only taken for the first VTOC subtask
that abends. If more than one subtask of this type abends,
the message is written to document which structure was being
processed, but the dump is bypassed.

See section 5.3.2 for instructions on what to do with the
dumps if you receive this message.
```

```
VCCVT15W
--------

TASK(nn) qqq Queue residuals  - Vol(vvvvvv) possible DSCB
         chain corruption

A VCCVTOC subtask completes processing for a VTOC and finds
that there are still unresolved entries in the DSCB
resolution queues.  This condition can occur if VCC is run
without VTOC serialization and an update of the VTOC occurs
during VCC processing, causing VCC to miss a DSCB chain
connection and resulting in unresolved entries in the
resolution queues.  There may actually be a broken DSCB chain
resulting from a hardware error.  If this message occurs, you
should take action to determine the extent of the problem.

  where:    qqq    = DSN if data set resolution queue or
                      EXT if free space resolution queue
            vvvvvv = Volume being processed at time of
                      abend

VCCVTOC action:  The subtask issuing this message continues
                 with the next VTOC.

VCCVT16E
--------

TASK(nn) SMF interface error  - Scan processing terminated

A VCCVTOC subtask receives a non-zero return code from the
VCCWRITR routine, and SMF was selected as the VCC recording
medium.

VCCVTOC action:  The subtask issuing this message terminates.


VCCVT17W
--------

TASK(nn) Unable to open VTOC  - Vol(vvvvvv)

A VCCVTOC subtask cannot open a VTOC.

VCCVTOC action:  The subtask issuing this message continues
                 with the next VTOC.
```

```
VCCVT18W
--------

TASK(nn) SMF record overflow  - Vol(vvvvvv) freespace extent
         segments incomplete

A VCCVTOC subtask is building the VTOC/VOLUME record and runs
out of room in the record to add more free space segments.
This is a warning message because volume free space segments
are not used for DASD space accounting.  The VCA reports for
the volume will not be valid.

VCCVTOC action:  The subtask issuing this message continues
                 with the next VTOC.

VCCVT19W
--------

TASK(nn) Lost tracks detected - Vol(vvvvvv) Lost(tttttttt)

  where:    vvvvvv   = Volume being addressed at time of
                       abend
            tttttttt = The number of tracks, in decimal, that
                       were unaccounted for by the scan task.

The verification calculation performed by the VTOC scan task
failed to reconcile the total number of tracks on the DASD
volume with the number of tracks that should exist based on
the physical geometry of the device.  There can be different
causes for this condition.  The most common cause is when
orphan extent DSCBs exist in the VTOC.

VCCVTOC action:  The subtask issuing this message continues
                 with the next VTOC.
```

```
VCCVT20E
--------

TASK(nn) Getmain failed for VSAM scan work area; Subtask
         terminated

The GETMAIN for the storage area that is used when data sets
of the form SYS1.VVDS.Vvolser are added to the CNQ fails.  If
you want to run with the number of VTOC scan subtasks that
were specified, increase the region size and rerun VCC.

VCCVTOC action:  The subtask issuing the error message
                 terminates.  All other subtask processing
                 continues.


VCCVT21W
--------

TASK(nn) Reserve retry failed - Vol(vvvvvv) reserve will not
         be done

Repeated attempts to reserve the VTOC for the indicated
volume have failed (the number of attempts is specified by
the RESRETRY run-time option).  Because RESFAILOK=Y was coded
or defaulted, the reserve is bypassed for this volume.

VCCVTOC action:  The subtask issuing the error message
                 will process the volume without doing a
                 reserve against it.  Data errors may result.

VCCVT22W
--------

TASK(nn) Reserve retry failed - Vol(vvvvvv) will not be
         processed

Repeated attempts to reserve the VTOC for the indicated
volume have failed (the number of attempts is specified by
the RESRETRY run-time option).  Because RESFAILOK=N was
coded, the volume will not be processed.

VCCVTOC action:  The subtask issuing the error message
                 will bypass this volume and select another
                 one to process.  The volume will not be
                 processed in this VCC run.
```

```
VCCVT23W
--------

TASK(nn) Storage group error  - Vol(vvvvvv) Ret(eeee)
         Rsn(rrrr)

where:

        vvvvvv  is the volume serial number for which storage
                group information was requested
        eeee    is the return code from the SMS address space
                in decimal
        rrrr    is the reason code associated with the SMS
                request and error

This message is produced when an attempt to get the storage
group name for a volume resulted in an error.

Processing continues with the name of the storage group set
to blanks for this volume.


VCCVT24W
--------

TASK(nn) Track calc error nnn - Vol(vvvvvv) Ret(eeee)
         Rsn(rrrr) defaulting to Device(3390)

where:
        nnn     is either "(P)" or "(D)" depending on whether
                the TRKCALC information was requested for the
                number of 4 KB pages per track (P) or the
                number of DSCBs per track (D).
        vvvvvv  is the volume serial number for which
                TRKCALC information was requested
        eeee    is the return code from the TRKCALC service
                routine
        rrrr    is the reason code associated with the
                TRKCALC error code

This message is produced when an attempt to get geometric
information about a DASD volume resulted in an error.

Processing continues with the information defaulted to a
DASD device type 3390.
```

```
VCCVT25I
--------

TASK(nn) VTOC collector initialization processing is
         complete; TCB(tttttt)

where:

        tttttt  is the virtual storage address in hex of the
                task control block (TCB) for the subtask

This message is produced following subtask initialization
completion for VTOC scanning.

VCCVTOC action:  The subtask begins VTOC scan processing.


VCCVT26W
--------

TASK(nn) Status check error   - Vol(vvvvvv) allocation will
         not be done

where:

        vvvvvv  is the volume serial number of the VTOC for
                which allocation was attempted

This message is produced whenever status indicators in the
UCB for the selected DASD volume indicate a condition that
will prevent successful allocation of the volume.

The queue entry for the selected volume is placed back on
the scan queue and the next available volume is selected.
```

```
VCCVT27W
--------

TASK(nn) Status check error   - Vol(vvvvvv) device is being
         bypassed

where:

        vvvvvv  is the volume serial number of the VTOC for
                which allocation was attempted

This message is produced whenever status indicators in the
UCB for the selected DASD volume indicate a condition that
will prevent successful allocation of the volume.

The queue entry for the selected volume is placed back on
the scan queue and the next available volume is selected.



VCCVT28I
--------

TASK(nn) VTOC collector fixed pp I/O buffer pages

where:

        pp  is the number of 4 KB I/O buffer pages that were
            fixed by the collector subtask

When conditions permit, the scan subtask will attempt to page
fix the I/O buffers to improve performance.

VCCVTOC action:  Processing continues normally. This is an
                 informational message.
```

```
VCCVT29W
--------

TASK(nn) DEVTYPE return error - Vol(vvvvvv) Ret(rr) Rsn(xx)

where:
        vvvvvv = Volume being processed at time of the
                 error return from the DEVTYPE service
        rr     = is the return code from the DEVTYPE service
        xx     = is the reason code associated with the
                 DEVTYPE service

VCCVTOC action:  The subtask issuing this message continues
                 with the next VTOC.


VCCVT30W
--------

TASK(nn) qqq Element(s) NTF   - Vol(vvvvvv) possible DSCB
         extent corruption

A VCCVTOC subtask was unable to locate a specified qqq
element on the qqq queue.  This is typically an extent entry
for a data set being processed.  This condition can occur if
VCC is run without VTOC serialization and an update of the
VTOC occurs during VCC processing, causing VCC to miss a DSCB
chain connection and resulting in unresolved entries in the
resolution queues.  There may actually be a broken DSCB chain
resulting from a hardware error.  If this message occurs, you
should take action to determine the extent of the problem.

  where:    qqq    = EXT if free space resolution queue
            vvvvvv = Volume being processed at time of
                     abend

VCCVTOC action:  The subtask issuing this message continues
                 with the next VTOC.
```

```
VCCVT31W
--------

TASK(nn) ABEND zzzz Recovery  - Vol(vvvvvv) Data Collection
         Not Initiated

A recoverable abend occurred while processing a VTOC. Since
the abend was determined to be recoverable, recovery is
invoked that performs cleanup on the environment and
releases any held resources.  The most common recoverable
abend is a S213, which occurs when attempting to open a VTOC
and there is no VTOC on the DASD volume.

  where:    zzzz   = System abend code in hex (Shhh) or
                     user abend code in decimal (Udddd)
            vvvvvv = Volume being processed at time of
                     abend

VCCVTOC action:  The subtask issuing this message continues
                 with the next VTOC.


VCCVT32E
--------

TASK(nn) ABEND zzzz Recursion - Vol(vvvvvv) Data Collection
         Not Initiated

An abend occurred during recovery processing of a previous
abend condition.  Regardless of whether the previous abend
was recoverable or not, the subsequent abend has rendered the
situation non-recoverable and the subtask will terminate.

  where:    zzzz   = System abend code in hex (Shhh) or
                     user abend code in decimal (Udddd)
            vvvvvv = Volume being processed at time of
                     abend

VCCVTOC action:  The subtask issuing this message is
                 terminated.
```

```
VCCVT34W   or   VCCCN34W
--------        --------

TASK(nn) Failure during UNPIN - Vol(vvvvvv) Device(dddd)
         Ret(eeee) Rsn(rrrr)

where:

        vvvvvv  is the volume serial number for which an
                attempt to UNPIN the UCB was requested.
        dddd    is the device number for which an attempt
                to UNPIN the UCB was requested.
        eeee    is the return code from the UCB services
                (in hex).
        rrrr    is the reason code associated with the UCB
                service request and failure (in hex).

These codes are documented in the appropriate IBM
publication.

Pinning a UCB ensures that the UCB cannot be deleted while a
program is in the process of looking at a UCB. VCCNTRL
establishes the PIN status for all online devices that will
be scanned, issuing the UNPIN service for devices to be
EXCLUDED. VCCVTOC issues the UNPIN service for devices that
scanned successfully; VCCVTOC also issues the UNPIN service
in the ESTAEX (error routine) for devices where the VTOC
scan did not complete successfully.

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.

VCCNTRL action:  Processing continues after the message to
                 VCCSTATS.
```

```
VCCVT40W
--------

TASK(nn) Allocation error     - Vol(vvvvvv) Ret(ee)
         Rsn(rrrrrrrr) DSN(dd)

where:

        vvvvvv  is the volume serial number of the DASD
                volume where the data set is located.
        eeee    is the return code from allocation
                services (in hex).
        rrrr    is the reason code from allocation services
                (in hex).
        dd      is the data set name of the data set being
                processed.

Prior to obtaining the extended attributes for a data set,
the data set is dynamically allocated. This message indicates
that an error occurred during this allocation.

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.  All of the normal processing for
                 the data set is performed, but none of the
                 extended attribute data is collected.  The
                 attributes not collected are those
                 concerning internal space utilization for
                 PDS/E data sets.
```

```
VCCVT41I
--------

TASK(nn) ssss uuuuuuuuuuuuuuu - Vol(vvvvvv) DSN(dd)

where:

        ssss    is the type of data set being processed
                (for example, "PDSE").
        uuuuuu  tells what is preventing the collection
                of extended attributes for the data set
                (for example, "Uncataloged").
        vvvvvv  is the volume serial number of the DASD
                volume where the data set is located.
        dd      is the data set name of the data set being
                processed.

The Data Set Attribute Retrieval (DSAR) function has
encountered a condition that prevents it from collecting the
extended attributes for the data set being processed.  In
this event, the extended attribute elements for the data set
being processed will contain zeros.

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.  All of the normal processing for
                 the data set is performed, but none of the
                 extended attribute data is collected.  The
                 attributes not collected are those
                 concerning internal space utilization for
                 PDS/E data sets.
```

```
VCCVT42I
--------

TASK(nn) ssss Error returned  - Ret(eeeeeeee) Rsn(rrrrrrrr)
         DSN(dd)

where:

        ssss      is the type of data set being processed.
                  (for example, "PDSE")
        eeeeeeee  is the return code value in hex that was
                  returned during DSAR processing.  In some
                  cases, the return code may have been
                  returned by an operating system service.
        rrrrrrrr  is the reason code value in hex that was
                  returned during DSAR processing.  In some
                  cases, the reason code may have been
                  returned by an operating system service.
        dd        is the data set name of the data set being
                  processed.

The Data Set Attribute Retrieval (DSAR) function has
encountered a condition that prevents it from collecting the
extended attributes for the data set being processed.  In
this event, the extended attribute elements for the data set
being processed will contain zeros.  This message may be
accompanied by 1 or more VCCVT47I messages that provide
additional descriptive information.

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.  All of the normal processing for
                 the data set is performed, but none of the
                 extended attribute data is collected.  The
                 attributes not collected are those
                 concerning internal space utilization for
                 PDS/E data sets.
```

```
VCCVT43E
--------

TASK(nn) ssss Attribute error - Field(aaaaaa) Ret(ee)
         DSN(dd)

where:

        ssss    is the type of data set being processed
                (for example, "PDSE").
        aaaaaa  is the name of the attribute element that was
                being processed by the DSAR service.
        ee      is the return code value in hex that was
                returned during DSAR processing.
        dd      is the data set name of the data set being
                processed.

The Data Set Attribute Retrieval (DSAR) function has
encountered an attribute specification error condition that
prevents it from collecting the extended attributes for the
data set being processed.  In this event, the extended
attribute elements for the data set being processed will
contain zeros.  This message may be accompanied by 1 or more
VCCVT47I messages that provide additional descriptive
information.

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.  All of the normal processing for
                 the data set is performed, but none of the
                 extended attribute data is collected.  The
                 attributes not collected are those
                 concerning internal space utilization for
                 PDS/E data sets.
```

```
VCCVT44E
--------

TASK(nn) ssss Missing data    - Field(aaaaaa) DSN(dd)

where:

        ssss    is the type of data set being processed
                (for example, "PDSE").
        aaaaaa  is the name of the attribute element that was
                being processed by the DSAR service.
        dd      is the data set name of the data set being
                processed.

The Data Set Attribute Retrieval (DSAR) function has received
incomplete extended attribute data from the operating system.
The design does not permit partial data element population.
In this event, all extended attribute elements for the data
set being processed will contain zeros.  This message may be
accompanied by 1 or more VCCVT47I messages that provide
additional descriptive information.

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.  All of the normal processing for
                 the data set is performed, but none of the
                 extended attribute data is collected.  The
                 attributes not collected are those
                 concerning internal space utilization for
                 PDS/E data sets.
```

```
VCCVT45E
--------

TASK(nn) ssss Unknown error   - Flags(xxxxxx) DSN(dd)

where:

        ssss    is the type of data set being processed
                (for example, "PDSE").
        xxxxxx  is an information field containing hex values
                indicating status of the error condition.
        dd      is the data set name of the data set being
                processed.

The caller of the Data Set Attribute Retrieval (DSAR)
function has been presented with a subordinate error code
that is not known to the caller.  The most likely cause of
this condition is a version mismatch of VCC modules in the
LOADLIB being used for the execution.  In this event, the
extended attribute elements for the data set being processed
will contain zeros.  This message may be accompanied by 1 or
more VCCVT47I messages that provide additional descriptive
information.

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.  All of the normal processing for
                 the data set is performed, but none of the
                 extended attribute data is collected.  The
                 attributes not collected are those
                 concerning internal space utilization for
                 PDS/E data sets.
```

```
VCCVT46E
--------

TASK(nn) ssss Unknown return  - Ret(eeeeee) DSN(dd)

where:

        ssss    is the type of data set being processed
                (for example, "PDSE").
        eeeeee  is the return code value in hex that was
                returned during DSAR processing.
        dd      is the data set name of the data set being
                processed.

The caller of the Data Set Attribute Retrieval (DSAR)
function has been presented with a return code that is not
known to the caller.  The most likely cause of this condition
is a version mismatch of VCC modules in the LOADLIB being
used for the execution.  In this event, the extended
attribute elements for the data set being processed will
contain zeros.  This message may be accompanied by 1 or more
VCCVT47I messages that provide additional descriptive
information.

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.  All of the normal processing for
                 the data set is performed, but none of the
                 extended attribute data is collected.  The
                 attributes not collected are those
                 concerning internal space utilization for
                 PDS/E data sets.
```

```
VCCVT47I
--------

TASK(nn) + variable_text

The Data Set Attribute Retrieval (DSAR) function has
encountered an exception condition as noted by a message in
the VCCVT41I thru VCCVT46E range.  This message contains
self-describing text that provides additional information
about the exception condition.

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.  All of the normal processing for
                 the data set is performed, but none of the
                 extended attribute data is collected.  The
                 attributes not collected are those
                 concerning internal space utilization for
                 PDS/E data sets.
```

```
VCCVT50W
--------

TASK(nn) ssss mmmmmmmmmmmmmmm - Vol(vvvvvv) Ret(rr) Rsn(xx)
         DSN(dd)

The Data Set Attribute Retrieval (DSAR) function was
attempting to process a PDS/E data set that is probably
damaged or corrupted.  An S0F4 abend has occurred within a
system service during this processing.  Additional
information will appear in a VCCVT51W scan log message.

where:
        ssss   = The type file that was being processed when
                 the error condition occurred; currently this
                 will either be PDSE or HFS
        mmmmmm = Description of the error condition that was
                 presented to the VTOC scan task; possible
                 descriptions include:

                   Info Returned
                   Warning
                   Invalid Parm
                   Caller Error
                   Environmental
                   Dir I/O Error
                   Media Error
                   Logical Error
                   Severe Error
                   Unknown Error

        vvvvvv = Volume being processed at time the error
                 condition occurred
        rr     = is the return code from the system service
        xx     = is the reason code associated with the
                 return code from the system service

VCCVTOC action:  Processing continues after the message to
                 VCCVTMSG.  All of the normal processing for
                 the data set is performed, but none of the
                 extended attribute data is collected.  The
                 attributes not collected are those
                 concerning internal space utilization for
                 PDS/E data sets.
```

## 7.2.8 VCCXHSM Messages

```
VCCHS02I
--------

PREVIOUS DATA COLLECTION (MIGRATE DATA)    yyddd   hh:mm:ss
                         (BACKUP  DATA)    yyddd   hh:mm:ss


The Julian date and time of day when this DFHSM data was
gathered previously.  The span of time between each inventory
measurement forms the DURATION data element in the CA MICS
Data Base that, in turn, is the basis for all "space*time"
calculations and a foundation for storage occupancy
accounting applications.  Usually, an inventory of the
migrate data will be taken at the same time as the inventory
of the backup data but because the HSMMIGRATE and HSMBACKUP
keywords allow you to vary the collection intervals of each
type of data, each previous time stamp is shown on the
report.

VCCXHSM  action:  These time stamps are obtained from the HSM
                  control record in the VCCNTROL file.  The
                  appropriate time stamp is stored in each
                  type 'M' and type'B' record written by
                  VCCXHSM to either SMF or to the VCCXHSM
                  data set.


VCCHS03I
--------

STARTING DATA COLLECTION:                  yyddd   hh:mm:ss


The Julian date and time of day when this DFHSM data
collection commenced.  A TIME macro is issued just prior to
calling ARCUTIL.

The span of time between this inventory and the previous time
forms the DURATION data element.

VCCXHSM  action:  This time stamp is stored in the HSM
                  control record in the VCCNTROL file. On the
                  subsequent run with HSM=Y, this time
                  becomes the PREVIOUS time stamp and a
                  perpetual recalculation of DURATION is
                  therefore possible.
```

```
VCCHS04I
--------

ENDING DATA COLLECTION:                    yyddd   hh:mm:ss


The Julian date and time of day when this DFHSM data
collection ended.  A TIME macro is issued upon return from
ARCUTIL.

This time stamp is shown for the purpose of understanding the
contribution to total VCC run-time caused by the DFHSM
inventory process.

VCCXHSM  action:  This time stamp is not stored in any file
                  nor placed in the VCCHSM records.  It is
                  displayed and used to calculate the elapsed
                  time shown in the VCCHS05I message.


VCCHS05I
--------

ELAPSED TIME FOR THIS RUN:   yyddd   hh:mm:ss


The calculated elapsed time spent in the VCCXHSM subtask
including the time required by ARCUTIL is shown.  It can be
compared to the elapsed time for the VTOC and catalog scans
done asynchronously by other functions of VCC.

VCCXHSM  action:  None.
```

```
VCCHS06I
--------

NUMBER OF TYPE "M" RECORDS IN --ML2-- READ:      nnnnnnnn
NUMBER OF TYPE "M" RECORDS IN --ML1-- READ:      pppppppp
NUMBER OF TYPE "M" RECORDS IN -SDSP-- READ:      ssssssss

The count and location of type "M" (migrate) records are
shown for informational purposes. Note that any records
found in a Small Data Set Packing (SDSP) data set are also
a part of ML1 (therefore the value for ssssssss above is
a subset of pppppppp). As an audit, the value of nnnnnnnn
+ pppppppp should equal the number of MIGRATE records
written as shown in the VCCHS07I message.

VCCXHSM action:  Messages produced at end of processing.

VCCHS07I
--------

NUMBER OF HSM -- tttttttt -- RECORDS PROCESSED:  nnnnnnnn

The type and count of records returned by ARCUTIL are shown
for auditing purposes.  Values for tttttttt are as follows:

  MIGRATE
  BACKUP
  DASD-CP
  TAPE-CP

VCCXHSM action:  Messages produced at end of processing.

VCCHS11E
--------

BLDL FAILED FOR ARCUTIL. HSM=N IS FORCED.

The required data collection program, ARCUTIL, could not be
found in the STEPLIB or LNKLST concatenation.  This would
suggest that either DFHSM is not installed at the required
version and release level or is installed in a STEPLIB that
is not in the concatenation with VCC.

VCCXHSM action:  Control is returned immediately to VCCNTRL
                 and VTOC and catalog scanning continues.
```

```
VCCHS12E
--------

ARCUTIL RETURN CODE = hh  REASON CODE = ii.

ARCUTIL detected some sort of environmental error and the
return code (hh) and the reason code (ii) are displayed in
decimal.  Refer to IBM publication SH35-0084-4, DFHSM
Installation and Customization Guide, for descriptions of
codes.  Your action should be guided by the return code:

RC   Action

0    Successful run; no action required.
4    Internal error in VCCXHSM; contact CA MICS Product
     Support.
8    Probable missing DD statement. See message VCCHS14E.
12   Probable hardware error. See message VCCHS15E.
16   Probable hardware error. See message VCCHS16E.
20   Abnormal termination in ARCUTIL.  See message
     VCCHS16E.
24   Internal processing error.  See messages VCCHS26E
     and VCCH27E.

VCCXHSM action:  Control is returned immediately to VCCNTRL
                 and VTOC and catalog scanning continues.

Problem Determination
---------------------

Most of the remaining messages attempt to supplement the
basic return code and reason code information in VCCHS12E.
The intent is to echo for your convenience what you would
find by examining the DFHSM Installation and Customization
Guide.

Direct Invocation of ARCUTIL
----------------------------

If an error occurs with the VCC interface to DFHSM, you
should run ARCUTIL directly to see if you get the same return
code.  If ARCUTIL results are the same as with the VCC
execution, you should follow appropriate debugging techniques
in IBM documentation for DFHSM and/or call IBM for technical
support.

To run ARCUTIL by itself, modify the following example as
needed:
```

```
//S1  EXEC PGM=ARCUTIL,PARM='DCOLLECT MIGD CAPD BACD MSERR'
//ARCSNAP DD SYSOUT=*
//ARCTEST DD SYSOUT=*
//ARCDATA DD DSN=prefix.ARCUTIL.BATCH.TEST,
//           DISP=(,CATLG,DELETE),UNIT=SYSALLDA,
//           SPACE=(CYL,(100,30),RLSE)
//MCDS    DD DSN=prefix.MCDS,DISP=SHR
//BCDS    DD DSN=prefix.BCDS,DISP=SHR
```

Because ARCUTIL does not compress output records like the VCC interface does, the space requirements for ARCDATA can be very significant.

More information about running ARCUTIL by itself can be found in Chapter 16 of the DFHSM Installation and Customization Guide.

VCCHS13E
--------

UNABLE TO OPEN DDNAME - VCCHSM

VCCXHSM requires this file to write data records unless HSMSMF=Y is coded.

If HSMSMF=Y is coded, then a small amount of space (perhaps a few tracks) is required as a work file.

VCCXHSM action:  Control is returned immediately to VCCNTRL and VTOC and catalog scanning continues.

VCCHS14E
--------

ERROR OPENING DFHSM CONTROL DATA SET. xxxx DDNAME IS MISSING OR INVALID.

ARCUTIL requires this file to obtain records from the xxxx (MCDS/BCDS) control data set.

VCCXHSM action:  Control is returned immediately to VCCNTRL and VTOC and catalog scanning continues.

```
VCCHS15E
--------

ERROR READING DFHSM xxxx DATA SET. OVER 1% OF RECORDS
REQUIRED CANNOT BE READ.

xxxx is substituted with MCDS or BCDS.

Investigate the cause of the READ errors and rerun the job.
Consider running ARCUTIL by itself as described by VCCHS12E.

VCCXHSM action:  Control is returned immediately to VCCNTRL
                 and VTOC and catalog scanning continues.


VCCHS16E
--------

ERROR WRITING TO 'VCCHSM'' DATA SET. REGISTER 1 CONTENT FROM
'SYNADAF' MACRO = hhhhhhhh.

An I/O error occurred writing DFHSM data to the output file
defined by the VCCHSM DD statement.  ARCUTIL has intercepted
this error by means of a SYNAD exit.

The content of register 1 associated with ARCUTIL error
handling is substituted for hhhhhhhh in the message.

Investigate the cause of the WRITE errors.  You may have to
reallocate the VCCHSM to a different and perhaps larger
location, then rerun the job.

Consider running ARCUTIL by itself as described by VCCHS12E.

VCCXHSM action:  Control is returned immediately to VCCNTRL
                 and VTOC and catalog scanning continues.
```

```
VCCHS17E
--------

ABNORMAL TERMINATION OCCURRED IN 'ARCUTIL'.
ABEND CODE = hhhhhhhh.

An abend has occurred within ARCUTIL processing. The abnormal
termination code is substituted for hhhhhhhh in the message
text.

Consider running ARCUTIL by itself as described by VCCHS12E.

If possible, correct the error and resubmit the job.  If this
is not possible, contact CA MICS Product Support and/or IBM
customer support.  You may be able to reproduce this error by
running DCOLLECT with an equivalent set of collection options
and JCL statements.


VCCXHSM action:  Control is returned immediately to VCCNTRL
                 and VTOC and catalog scanning continues.
```

```
VCCHS18E
--------

ABNORMAL TERMINATION OCCURRED DURING HSM=Y PROCESSING.
ABEND COMPLETION CODE = hhhhhh

An abend has occurred within VCCXHSM or possibly within
ARCUTIL.  The abend code is placed in the message.

Consider running ARCUTIL by itself as described by VCCHS12E.

Contact CA MICS Product Support. If possible, have the dump
on SYSUDUMP available.

VCCXHSM action:  Control is returned immediately to VCCNTRL
                 and VTOC and catalog scanning continues.

VCCHS22I
--------

SKIPPING RECORD TYPE 'x' DUE TO INVALID DATA. ID = dsname

A validity check on the length of the record (to avoid abend
S002-18) has trapped an invalid record that will be skipped
(not written to the output destination). If it is a type 'M'
or type 'B' record, the data set name will be placed in the
message to help identify this error.

VCCXHSM action:  The record is skipped and the skip counter
                 is updated.  The number of skipped records
                 is reported at the end of the job.


VCCHS23I
--------

NUMBER OF RECORDS SKIPPED BY THIS INTERFACE: zzzzzzz

The count of records found to have an invalid record length
or other detectable error is reported.

VCCXHSM action:  None.
```

```
VCCHS25E
--------

ERROR READING DFHSM CONTROL DATA SET. POSITION ERROR TO
xxxx RECORD OCCURRED.

xxxx is substituted with MCDS or BCDS.

Consider running ARCUTIL by itself as described by VCCHS12E.

VCCXHSM action:  Control is returned immediately to VCCNTRL
                 and VTOC and catalog scanning continues.

VCCHS26E
--------

INTERNAL ERROR. ARCSNAP DD NOT AVAILABLE.

Consider running ARCUTIL by itself as described by VCCHS12E.

VCCXHSM action:  Control is returned immediately to VCCNTRL
                 and VTOC and catalog scanning continues.


VCCHS27E
--------

INTERNAL ERROR. ESTAE CANNOT BE ESTABLISHED.

Consider running ARCUTIL by itself as described by VCCHS12E.

VCCXHSM action:  Control is returned immediately to VCCNTRL
                 and VTOC and catalog scanning continues.
```

```
VCCHS28I
--------

NUMBER OF TYPE "M" RECORDS ON ML2 SKIPPED:     nnnnnnnn

This message refers to a "filter" that occurs when VCC is
executing in "ASTEX mode" only. All type "M" records for data
sets migrated to ML2 before the implementation date of the
ASTEX Migration Manager are skipped and a count of those
records thus filtered out is shown in this message as
nnnnnnnn.  This filter is to improve performance since the
records would have been discarded later by Migration Manager
post-processor steps.

VCCXHSM action:  None.
```

## 7.2.9 VCCMAIN Messages

```
VCC0001I  variable_text
--------

This is an informational message issued to the System
Console. No response is necessary.  VCC processing continues.


VCC0002I  variable_text
--------

This is an informational message with no response necessary.
VCC processing continues.


VCC0003I  variable_text
--------

This is an informational message with no response necessary.
VCC processing continues.


VCC0005I  Console operators reply was: xxxxxx
--------

This is an informational message indicating the System
Console Operators response to an action message.  VCC
processing continues.
```

```
VCC0006E  Invalid reply - xxxxxx
--------

'xxxxx' in the above message indicates the incorrect response
the System Console Operator used in responding to an action
message.  View the message log to find the action message
that was issued prior to this one.  Make sure the response is
among the list of valid responses shown for the action
message.  VCC processing continues.

VCC0007E  Invalid parameter - xxxxxx
--------

'xxxxx' in the above message indicates the incorrect value
specified for a VCC parameter. Correct the value and resubmit
the VCC run.
```

```
VCC0008E  Invalid keyword - xxxxxx
--------

'xxxxx' in the above message indicates the incorrect VCC
parameter specified. Correct the value and resubmit the VCC
run.


VCC0009E  Unable to continue processing - reason_text
--------

This message signals that VCC encountered a problem during
processing and was unable to recover.  'reason_text'
identifies the cause of the problem.  Processing is
terminated.  Research the cause and rerun VCC.  the following
are possible values for 'reason_text':


NOT ENOUGH STRG - VCC was unable to continue process because
                  insufficient virtual main storage was
                  allocated. Increase the size on the REGION
                  JCL parameter.

NOT AUTHORIZED - VCC is not running as an authorized program.
                 Make sure that the VCC modules are located
                 in an authorized library. If a JOBLIB or
                 STEPLIB DD is used, make sure that all other
                 concatenated libraries are also authorized.
                 The z/OS restriction that any unauthorized
                 library in an authorized JOBLIB/STEPLIB
                 concatenation forces all libraries to be
                 treated as unauthorized may have been in
                 effect in this case.

VOLTBL FULL - The DASD volume table allocated by VCC has
              reached its limit. Restart VCC.
```

```
VCC0010I  Processing completed, Return Code nn
--------

This message signals that VCC processing has completed. 'nn'
is the return code.


VCC0011E  Required DD statement xxxxxxx not specified
--------

This message is issued when VCC detects that a required DD
JCL statement was not specified. 'xxxxxxx' indicates the
DDname. Add the correct JCL statement and resubmit the VCC
run.

VCC0012I  CA MICS Space Collector Initialization in Progress
--------

This message signals that VCC initialization processing is in
progress.

VCC0013I  CA MICS Space Collector Initialization Complete
--------

This message signals that VCC initialization processing has
ended successfully. VCC is now ready for full processing.


VCC0014I  nnn,nnn,nnn records written to dsname on volume
--------  volser

This message indicates the number of records written to the
VCC output data set. 'dsname' is the output data set name;
'volser' is the volume serial location of the output data
set.
```

```
VCC0015I  Recording data on volume volser, Dataset: dsn
--------

This message is issued when VCC has started writing records
to the output data set. 'volser' is the volume serial
location of the output data set; 'dsn' is the output data set
name.


VCC0016E  I/O error on file ddn - desc
--------

This message indicates that an error occurred while VCC was
reading or writing to the data set referenced by DDname
'ddn'.  'desc' is the description of the actual error
encountered.  Correct the problem and rerun VCC.

VCC0017I  Read USS File Systems Subtask nn
--------  Started|Suspended|Ended

This message indicates when the various types of subtasks
within VCC start or complete processing. 'xxxxx' is the
subtask type, such as READVTOC, READCTLG, READHFS. 'nn' is
the sequence number of the subtask. The two digit subtask
identifier is located right after the timestamp for every
message issued by a subtask within the message log data set
(SYSPRINT).
Informational only.
```

```
VCC0018I  Now scanning xxxxxxx yyyyyyy
--------

This message indicates when a VCC subtask has started to
process a file system, DASD Volume, or System Catalog.
'yyyyyy' is the entity name such as the volume serial or
data set name. 'xxxxxx' is the entity type identifying what
the entity is. This message is informational only.


VCC0019E  modname was unable to obtain nnnnnK
--------

The indicated VCC module was unable to obtain enough virtual
storage for its processing. Increase the region size at least
by the amount shown and rerun VCC.


VCC0020I  Set VolumeTableLength(nnnnK)
--------

This message indicates the amount of storage allocated in
units of 'k' (1,024 bytes) by VCC in building its internal
DASD Volume Table.

VCC0021I  nn,nnn Mounted file systems found
--------

This message indicates the number of Hierarchical File
Systems mounted on z/OS that VCC found and will process.

VCC0022I  nn,nnn online DASD volumes found
--------

This message indicates the number of Direct Access Storage
Devices online and available to VCC that will be processed.
```

```
VCC0023E  I/O error on unit xxxx Volume: vvvvvv - EC: zzzz
--------

VCC issues this message when it has been unable to
communicate successfully with the DASD Storage Subsystem or
Control unit that volume 'vvvvvv' is attached to.  'xxxx'
indicates the device address.  'zzzz' is diagnostic
information returned in the Channel Status Word (CSW) to aid
CA MICS Product Support personnel in determining the cause of
the problem.

Processing continues for the next volume.  Some information
regarding the device that contained the error may not be
available in the VCCDATA record.

VCC0024I  Diagnostic trace requested for unit xxxx vol:vvvvv
--------

VCC issues this message when an error has been detected while
trying to access information from the Control Unit for the
indicated volume (vvvvvv), and the TRACE=Y parameter was
specified.  'xxxx' indicates the device address.  The
information provided assists CA MICS Product Support
personnel in determining the cause of the problem.  Also, see
VCC0023E.

VCC0025W  Device information unavailable for volume vvvvv on
--------  unit zzzz

VCC issues this message when a DASD device is not in the
ready state or VCC could not communicate successfully with
the DASD Storage Subsystem or Control unit that volume
'vvvvvv' is attached to.  'zzzz' indicates the device
address.  Determine why the device was unavailable during the
VCC run.  The indicated volume will not be processed by VCC.
Processing continues for the next volume. Return code 4 is
set.
```

```
VCC0026W  nnnn Warning|Error message(s) were issued. Please
--------  check job output.

This message is issued to the System Console during
termination processing indicating that warning and/or error
messages were issued during VCC processing. This message
helps to alert the personnel responsible for checking for a
successful VCC run.


VCC0027W  Insufficient authorization to scan file systems.
--------

This message is issued when VCC does not have the required
access level to scan a file system. Ensure that VCC has
authority to use the resource type IBMFAC with the resource
BPX.DAEMON.

VCC0028E  Run required backup for output file: ddname
--------  Dataset: dsn

This message is issued when VCC detected that the previous
output data set was not copied successfully. Refer to the
BYPASSBKUP runtime parameter described earlier within this
document.


VCC0030W  No ddname DD statement specified, Default values
--------  in effect.

VCC could not open the VCCPARMS DD statement. VCC will use
its default values. VCC processing continues.
```

```
VCC0031I  Input parameter file ddname dsn on volume volser
--------

This informational message identifies where VCC obtained its
runtime parameters.  Processing continues.


VCC0032E  Missing equals sign
--------

A parameter keyword has been identified but is not
immediately preceded by an equals sign (with no intervening
blanks).  Parameter keyword parsing continues.  Return code 8
is set.


VCC0033E  Keyword value must be "Y" or "N"
--------

A parameter keyword is encountered whose only valid values
are Y (yes) and N (no), and one of these values was not
entered.  Parameter keyword parsing continues.  Return code 8
is set.

VCC0034E  Keyword value must be numeric
--------

A parameter keyword requiring numeric values contains a value
that is not numeric.  Parameter keyword parsing continues.
Return code 8 is set.


VCC0036E  Keyword value too large
--------

A parameter keyword requiring numeric values was encountered,
but the numeric value entered is too large for VCC to handle.
Parameter keyword parsing continues.  Return code 8 is set.
```

```
VCC0037E  Keyword value out of allowable range
--------

A parameter keyword is encountered whose only valid options
are numeric values and the numeric value entered is outside
of the range of values allowed for this keyword.  Parameter
keyword parsing continues.  Return code 8 is set.

VCC0038E  Invalid keyword value
--------

A parameter keyword is encountered that has a list of valid
numeric or character values associated with it, but the value
entered is not in this list.  Parameter keyword parsing
continues.  Return code 8 is set.


VCC0039E  Keyword specification too long
--------

A parameter keyword value was encountered that has a maximum
length that was exceeded.  Parameter keyword parsing
continues.  Return code 8 is set.

VCC0040E  Invalid data set name
--------

A parameter keyword is encountered whose value must be data
set names and one of the values specified is not a valid data
set name.  Parameter keyword parsing continues.  Return code
8 is set.


VCC0041E  Missing left|right parenthesis
--------

A parameter keyword is encountered whose values must be
enclosed in parentheses and the left or right parenthesis is
missing.  Parameter keyword parsing continues.  Return code 8
is set.

VCC0043E  Keyword terminator must be blank
--------

A parameter keyword is encountered that can have only one
option and the option specified does not end in a blank
character.  Parameter keyword parsing continues.  Return code
8 is set.
```

```
VCC0044E  Missing keyword value
--------

A parameter keyword is encountered that has no value
specified after the equals sign and no intervening blanks.
Parameter keyword parsing continues.  Return code 8 is set.

VCC0045E  Invalid list separator character
--------

A parameter keyword is encountered whose value must be a data
set name or other alphanumeric value, yet does not end with a
comma, period, or right parenthesis.  Parameter keyword
parsing continues.  Return code 8 is set.

VCC0046E  Missing terminator
--------

A parameter keyword is encountered whose value does not
terminate on the same card.  For keywords with a list of
options, one of the items in the list does not terminate on
the same card on which the options began.  Parameter keyword
parsing continues.  Return code 8 is set.

VCC0047E  Keyword mutually exclusive with another
--------

A parameter keyword is encountered whose mutually exclusive
counterpart has been encountered previously.  This applies to
all INCLUDE/EXCLUDE parameter keywords.  Parameter keyword
parsing continues.  Return code 8 is set.

VCC0048E  Generic specification not allowed for this keyword
--------

A parameter keyword is encountered whose operand contains an
asterisk.  Parameter keyword parsing continues.  Return code
8 is set.
```

```
VCC0049E  Maximum entries for keyword exceeded nnn
--------

A parameter keyword is encountered whose options are a list
of values, and the number of entries specified in this list
exceeds the maximum allowed for the keyword. The string nnn
in the message text is replaced by the maximum number of
entries defined for the parameter keyword currently being
parsed.  Parameter keyword parsing continues.  Return code 8
is set.

VCC0050W  Keyword is obsolete and should be removed
--------

A parameter keyword is encountered that has been obsoleted
and VCC no longer uses.  The parameter keyword and its value
are ignored.  Parameter keyword parsing continues. Return
code 4 is set.

VCC0051W  Incorrect keyword value; xxxxx=yyyyy in effect
--------

The parameter value is insufficient or conflicts with the
value of a previously specified keyword. VCC will force the
value shown by 'xxxxx=yyyyyy'. Processing continues and
return code 4 is set.

VCC0052E  Keyword value conflict - xxxxxx
--------
VCC detected a conflict with the indicated parameter 'xxxxx'
and another parameter specification. VCC is terminated.
Resolve the conflict and rerun VCC.

VCC0053E  Expected continuation not received
--------
VCC was expecting an operand of a runtime parameter to
continue onto the next statement but did not find it.
Respecify or add the missing operand and rerun VCC.


VCC0054W  Entry ignored - duplicate specification
--------

VCC detected a runtime parameter that was already specified
previously. The parameter shown prior to this message is
ignored.  Remove the duplicate parameter if desired.
```

```
VCC0059W  Unable to write output record type nnn - RSN=zz
--------

VCC was unable to write the output record to the data set
referenced by the VCCDATA DD statement or to the SMF z/OS
subsystem. 'zz' indicates the reason code for determining the
cause. The following is a list of causes and their associated
reason codes:

Reason code   Description
-----------   -------------------------------------------
    1         VCC SMF RECORD RDW IS LESS THAN 18 BYTES.  If a
              user exit is being used, make sure that the
              exit is not creating records less than 18 bytes
              in length. Otherwise, this is a problem that
              should be reported to CA MICS Product Support.

    2         VCC SMF RECORD RDW IS GREATER THAN 4096 BYTES.
              If a user exit is being used, make sure that
              the exit is not creating records greater than
              4K in length. Otherwise, this is a problem that
              should be reported to CA MICS Product Support.

    6         SMF IS NOT ACTIVE.  Report this message to your
              systems programmer.

    7         SMF EXIT IS REJECTING VCC SMF RECORDS
              SMF Exit IEFU83 is rejecting SMF records with
              the record type 'nnn'.  See your systems
              programmer to ascertain if you are using the
              proper SMF record type.

    8         VCC SMF DATA RECORD HAS BEEN LOST
              All z/OS SMF data logging data sets are full so
              no more SMF records can be written.  Report
              this problem to your systems programmer.

    9         SMF BUFFER SHORTAGE
              SMF does not have enough buffers to handle the
              record output rate currently being experienced
              by the system.  Inform your systems programmer.

   10         SMFWTM SEVERE ERROR
              Report this problem to your systems programmer.

VCC will try to continue processing. If unsuccessful, or the
problem occurs again, VCC will terminate.
```

```
VCC0060W  SMF recording is suppressed for record type nnn
--------

VCC detected that SMF record type 'nnn' is not being recorded
by the SMF subsystem. VCC will terminate processing.  Report
this problem to your systems programmer.

VCC0061W  xxxxxxxx Request failed RC=yy, RSN=zzzz pppp
--------

VCC was unsuccessful in performing the request indicated by
'xxxxxx'. The return ('yy') and reason codes ('zzzz') in the
message text are provided to assist in determining the cause
of the problem.  VCC processing will continue.  Function
names that begin with 'USS' are z/OS UNIX System Services
requests. The POSIX path name causing the error ('pppp') may
be shown in the message text.  You make look up the return
and reason codes in the z/OS UNIX System Services Messages
and Codes documentation to determine the cause of the
failure.  Try to correct the problem. Otherwise report this
message to CA MICS Product Support.

VCC0062E  xxxxxxxx Request failed RC=yy, RSN=zzzz pppp
--------

VCC was unsuccessful in performing the required request
indicated by 'xxxxxx' 'xx' and 'yy' (return and reason codes
to assist in determining the cause of the problem).  VCC
processing will terminate.  Report this message to CA MICS
Product Support.

VCC0063E  Abend xxxxx detected by mod in yyyyy
--------

VCC's ESTAE routine received control due to an abnormal
condition.  'xxxxx' indicates the abend code. 'mod' is the
VCC module that detected the error. 'yyyy' is the CSECT or
subroutine name.   VCC will issue an SVCDUMP to the system
dump data sets and terminate. Save the dump and report this
message to CA MICS Product Support.


VCC0064I  Diagnostic dump request of xxxxxx started|ended
--------

VCC issues this informational message indicating the status
of the request to dump VCC's virtual storage. Informational.
```

```
VCC0065I  Collection shutdown requested
--------

When this message is issued VCC has obtained a request from
the System Console Operator to terminate processing.
This message is informational only.


VCC0066W  xxxxxx request failed yyyyyyy
--------

VCC was unsuccessful in performing the request indicated by
'xxxxxx', but was able to continue processing. 'yyyy'
indicates the action taken by VCC so that it could continue
processing.  You may review the cause of the problem and
correct it for a future execution of VCC.

VCC0070I  Control Data Set dsn is on volume vvvvvv
--------

VCC issues this informational message when its control data
set is opened. VCC Processing continues.

VCC0071A  Control Data Set in use by xxxxx on system yyyyy
--------  Reply: 'W'-Wait or 'C'-Cancel

VCC issues this message to the System Console when it detects
that the control data set is in use for an unusually long
period of time by another job. 'xxxxx' is the jobname, and
'yyyyy' is the name of the system that 'xxxxx' is executing
on.  The Console Operator should determine if the indicated
job is running successfully. If this is the case, reply 'W'
to wait for the job. Otherwise reply 'C' to cancel this VCC
run. Rerun VCC at a later time.
```

```
VCC0072A  Control Data Set in use by xxxxx on system yyyyy
--------  Reply: 'U'-Reset/Use, 'W'-Wait, or 'C'-Cancel

VCC issues this message to the System Console when it detects
that the control data set is in use for an unusually long
period of time by another job. 'xxxxx' is the jobname, and
'yyyyy' is the name of the system that 'xxxxx' is executing
on.  The Console Operator should determine if the indicated
job is running successfully. If this is the case, reply 'W'
to wait for the job. If job 'xxxxx' is not currently
executing, reply 'U' to reset the control file. Most likely
the previous VCC run using the control file ended abnormally
leaving the control file in a locked condition. If the reply
is 'C' VCC will be terminated.

VCC0073E  Insufficient space for Control Data Set
--------

VCC issues this message when it detects that the control data
set allocated does not contain sufficient space needed by
VCC.  VCC will terminate. Reallocate the control data set
with more space and rerun VCC.

VCC0074E  Control Data Set is corrupted
--------

VCC detected that the control data set has been overwritten
with other unknown information. A dump will be written to the
data set referenced by the VCCSNAP DD statement.  Save the
dump and Report this message to CA MICS Product Support.  VCC
processing terminates.

VCC0075E  Control Data Set integrity exposure
--------

VCC detected that the control data set has lost its data
integrity due to another VCC job running on another system.
This problem may occur when DASD RESERVES have been converted
to ENQUEUES by a system product.  VCC processing terminates.
A dump will be written to the data set referenced by the
VCCSNAP DD statement.  Save the dump and report this message
to CA MICS Product Support.
```

```
VCC0077A  Control Data Set in use by xxxxx
--------  Reply: 'U'-Reset/Use, 'W'-Wait, or 'C'-Cancel

VCC issues this message to the System Console when it detects
that the control data set is in use for an unusually long
period of time by another VCC Job named 'xxxxxx'.  The
Console Operator should determine if the indicated job is
running successfully. If this is the case, reply 'W' to wait
for the job. If job 'xxxxx' is not currently executing reply
'U' to reset the control file. Most likely the previous VCC
run using the control file ended abnormally leaving the
control file in a locked condition. If the reply is 'C' VCC
will be terminated.

VCC0078I  Control Data Set initialized|converted
--------

VCC issues this informational message when the control data
set has been either converted from its old format or is a new
data set that has been initialized successfully.  VCC
processing continues.

VCC0080I  xxxxxx entry yyyyy expired, removed from control
--------  file

While writing necessary information regarding the devices
and/or file systems scanned, VCC found an entry that has
expired and has removed the entry from the control file.
'xxxxxx' indicates the entry type such as volume, file
system, or system catalog. 'yyyyy' indicates the entry name.
This message is informational. VCC processing continues.
```

```
VCC0081I  Control Data Set extended by nnnk
--------

During processing VCC has detected that the control data set
is too small to store necessary information regarding the
devices and/or file systems scanned. This message indicates
that VCC has successfully extended the size of the control
data set by the number of bytes shown. VCC processing will
continue.


VCC0084I  nnnn xxxxx were already processed
--------

This message is issued when VCC has detected that some
entities were already processed previously by VCC. Hence the
entities, either volumes, file systems or system catalogs,
will not be processed during this collection run. 'nnnn'
indicates the number of entities that will not be processed.
'xxxxx' indicates the entity type.
This message is informational. VCC processing continues.

VCC0085I  variable_text
--------

This is an informational message issued to the System
Console in response to an operator command. No response
is necessary.  VCC processing continues.

VCC0089I  Set ReadDirectoryLevels(nnn)
--------

This is an informational message issued to report the number
of file system directory read levels that are being used for
the USS FIle System scan execution. This value is either
specified or defaulted by the HFSDIRLEVEL parameter.  No
response is necessary.  VCC processing continues.

VCC0090I  Set DirectoryBufferSize(nnK)
--------

This is an informational message issued to report the size of
the file sYstem Directory Read Buffers that are being used
for the USS File System scan execution. This value is either
specified or defaulted by the HFSDIRBUFFER parameter.  No
response is necessary.  VCC processing continues.
```

```
VCC0091I  Group|User ID Build Processed nnnn Definitions
--------

This is an informational message issued to report on the
number of Group or User ID definitions that were read from
the USS Group or User database.  No response is necessary.
VCC processing continues.

VCC0092I  Group|User ID Build Created nnnn Table Entries
--------

This is an informational message issued to report on the
number of Group or User ID entries that were created in the
storage table used during scan processing. This number will
almost always be less than the corresponding number reported
in the VCC0091I message due to the elimination of duplicates.
No response is necessary.  VCC processing continues.


VCC0093I  Group|User ID Build Released nnnK of Unused Storage
--------

This is an informational message issued that indicates how
much unused virtual storage was released after completion of
the Group or User ID table build process.  No response is
necessary.  VCC processing continues.

VCC0094E  Group|User ID Build Maximum Table Entries Exceeded
--------

This is an error message issued to indicate that either the
Group ID or the User ID table has overflowed. It is unlikely
that this condition will be raised. This is due to the design
which allocates extremely large virtual storage areas for the
tables then releases the unused portions. The default size
for each of the tables is adequate to contain 32,767 entries.
No response is necessary.  VCC processing continues.

VCC0100I  File Systems Processed.........nnnn
--------

This is an informational message issued to report on the
number of files systems that the identified subtask
processed.  No response is necessary.  VCC processing
continues.
```

## 7.2.10 - VCCDIAG Messages

```
VCCXD00I
--------

(This is a blank line used for message formatting.)

Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD01E
--------

Scan Task Diagnostics   - Abend ZZZZ Rsn(RRRRRRRR)
  Task(TTTTTTTT) Func(FFFFFFFF)

This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.


Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD02I
--------

Active Task Information - ASID(AAAA) ASCB(BBBBBBBB)
  TCB(TTTTTTTT)

This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.
```

```
Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD03I
--------

Module(MMMMMMMM) Length(LLLLLLLL) Section(SSSSSSSS)
  PID(PPPP) CSV(QQ) NLU(NN)

This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.

Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD04I
--------

Load(LLLLLLLL) Entry(EEEEEEEE) PDA(PPPPPPPP) DSA(DDDDDDDD)
  BEA(XXXXXXXX_YYYYYYYY)

This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.

Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.
```

```
VCCXD05I
--------

Request Block (PRB ) Information
              -or-
              SVRB

This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.

Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD06I
--------

Module(MMMMMMMM) Section(SSSSSSSS) PID(PPPP) CSV(QQ) NLU(NN)

This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.

Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD07I
--------

Load(AAAAAAAA) Length(LLLLLLLL) Entry(EEEEEEEE)
  Offset(OOOOOOOO)
```

```
This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.

Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD08I
--------

PSW(PPPPPPPP QQQQQQQQ) ILC(L) INT(II) ZZZZ(XXXXXXXX)
  SSSSSSSSSSSSSSSSSSSSS

This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.

Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD09I
--------

Data at Address(AAAAAAAA) IIIIIIII IIIIIIII IIIIIIII

This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.
```

```
Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD10I
--------

RSS - REE: RRRRRRRR RRRRRRRR RRRRRRRR RRRRRRRR

This message is one of a set of messages that are issued by
the Extended Diagnostics processing routine. The messages do
not contain any information that is useful to the user.  The
information produced is meant to be used by the Customer
Support personnel for diagnosing and debugging an error
condition that has occured.

Action:  Either recovery or termination processing proceeds
         depending upon the abnormal condition encountered.
         The output data set(s) containing this message
         (VCCSTATS, VCCVTMSG, VCCVSMSG, or VCCHSMSG) should
         be kept for submission to Customer Support if
         necessary.


VCCXD11I
--------

Dump has been scheduled; Keep SYS1.DUMPnn data set
  for analysis

This message is issued as a result of an error condition
which caused the scheduling of and SVC dump.

See Section 5.3.2 for specifics on handling SYS1.DUMPnn
data sets.

Action:  Recovery termination processing continues.


VCCXD12E
--------

DASD Space Collector Abend ZZZZZRsn(RRRRRRRR)
  Task(MMMMMMMM) Func(FFFFFFFF)
```

This message is produced as the dump header for a SYS1.DUMPnn
created in response to an error condition.

Action:  Recovery termination processing continues.