

# CA MICS® Resource Management

## Space Analyzer Option Guide

Release 12.9



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

## 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](mailto: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

---

<b>Chapter 1: OVERVIEW</b>	<b>9</b>
1.1 Primary Areas of Application.....	10
1.2 Reporting and Inquiry Facilities.....	14
1.3 Requirements.....	14
<b>Chapter 2: USAGE GUIDELINES</b>	<b>15</b>
2.1 Methods of Using the Data .....	16
2.1.1 Using SAS for Retrieval and Reporting .....	17
2.1.2 Overview of the CA MICS Information Center Facility .....	17
2.1.3 Major Features of MICF .....	19
2.2 Management Support Applications .....	23
2.2.1 CA MICS Accounting and Chargeback .....	24
2.2.2 CA MICS Capacity Planner .....	25
2.2.3 CA MICS StorageMate .....	26
2.3 Miscellaneous Usage Notes for All Files.....	27
2.3.1 The Inventory Nature of the Data .....	27
2.3.2 The Calculation and Importance of DURATION .....	29
2.3.3 Detail and Summary Data Considerations .....	30
<b>Chapter 3: REPORTS</b>	<b>33</b>
3.1 DASD Installation Overview Report.....	36
3.2 Account Summary Report .....	40
3.3 Volume Summary Report .....	42
3.4 Detail Data Set Report.....	43
3.5 Detail VSAM Data Set Report .....	45
3.6 Volume Track Map Report .....	47
3.7 HFS Detail File/Directory Report .....	49
3.8 Using the MICF Standard Analysis Reports .....	50
3.8.1 Executing the MICF Standard Analysis Inquiries .....	50
3.8.2 Execution Time Override Options .....	52
3.8.3 Macro Variable Override Options .....	71
3.8.4 Sample Inquiry Execution.....	72
3.8.5 MICF Production Reporting Support.....	76

---

## Chapter 4: EXCEPTIONS 77

4.1 Exception Process Overview .....	78
4.2 Setting Exception Values .....	81
4.3 Detailed Exception Descriptions.....	82
09001: Available VTOC DSCB Shortage .....	83
09002: Available Volume Space Shortage.....	84
09003: High Alternate Track Usage.....	85
09004: Volume Free Space is Fragmented.....	87
09005: Data Set has Too Many Extents .....	88
09006: VSAM Data Set has Too Many Extents.....	89
09007: Fixed Length PO or PS Data Set is Unblocked .....	90
09008: DA Data Set is in Multiple Extents.....	91
09010: Data Set Using Less than n% of Allocated Space.....	92
09012: VTOC Reorganization Recommended .....	93
09013: VTOC in Error, Requires Restructuring.....	95
09014: Tracks Not Accounted for in VTOC.....	96
09015: VTOC Does Not End on a Cylinder Boundary .....	97
09016: Large Data Set Not Cylinder Allocated .....	98
09017: VTOC is Not in Indexed Format .....	99
09018: Indexed VTOC Potentially in Error.....	101
09019: Data Set Has An Invalid DS1LSTAR Value .....	102

## Chapter 5: FILES 103

5.1 Data Element Naming Conventions .....	105
5.2 VTOC/Catalog Activity Information Area Files .....	106
5.2.1 Data Set Allocation File (VCADAA) .....	107
5.2.2 VSAM Data Set Allocation File (VCA_VS).....	119
5.2.3 Volume Allocation File (VCAVOA) .....	133
5.2.4 Data Class Summary File (VCADCS).....	143
5.3 DFHSM Inventory Primary Information Area Files .....	148
5.3.1 Migrated Data File (HSMMIG).....	149
5.3.2 Volume Planning File (HSMVOL) .....	155
5.3.3 Tape Planning File (HSMTAP) .....	165
5.4 DFHSM Inventory Backup Information Area Files .....	170
5.4.1 BACKUP Data File (HSBBAC) .....	171
5.5 USS File System Information Area File .....	179
5.5.1 USS Directory Entry File (HFSDIR) .....	179

## Chapter 6: DATA SOURCES 187

6.1 CA MICS and CA SMF Director Interface .....	187
---	-----

---

## **Chapter 7: PARAMETERS** **189**

7.1 Environmental Considerations .....	190
7.2 Complex Level Parameters .....	190
7.2.1 Define Account Code Structure (VCAACCT) .....	191
7.2.2 Account Code Derivation Exit (VCAACRT) .....	195
7.2.3 Synchronize HSMACCT with VCAACCT .....	201
7.2.4 Synchronize HSMACRT with VCAACRT .....	203
7.2.5 Analyzer Definition Statements (VCAGENIN) .....	205
7.3 Unit Level Parameters .....	207
7.3.1 VCA Processing Options (VCAOPS) .....	209
7.3.2 HSM Processing Options (HSMOPS) .....	250
7.3.3 INPUTRDR, INPUTVCA, and INPUTHSM PARMS Members .....	288
7.3.4 Database Space Modeling (DBMODEL) .....	294

## **Chapter 8: INSTALLATION** **297**

## **Chapter 9: PROCESSING** **299**

9.1 Processing Overview .....	300
9.1.1 Relationship to VCC SMF Record Production .....	301
9.1.2 Updating the VCA Information Area .....	302
9.1.3 Updating the HSM and HSB Information Areas .....	304
9.2 Daily Processing Flow for VCA (DAY090) .....	306
9.2.1 MICSLOG Reporting .....	307
9.2.2 Phase 1 - Input Raw VCC and HFS SMF Data .....	310
9.2.3 Phase 2 - Sorts .....	311
9.2.4 Phase 3 - Match VOA to Catalog .....	311
9.2.5 Phase 4 - Merge Catalog Information .....	312
9.2.6 Phase 5 - Merge VSAM Information into VOA .....	315
9.2.7 Phase 6 - Match VOA to DAA .....	316
9.2.8 Phase 7 - Summarization and Aging .....	318
9.3 Daily Processing Flow for HSM and HSB (DAY095) .....	319
9.3.1 MICSLOG Reporting .....	320
9.3.2 Phase 1 - Input Raw VCC SMF Data .....	322
9.3.3 Phase 2 - Sort and Build Files .....	324
9.3.4 Phase 3 - Summarization and Aging .....	326
9.3.5 Description of Phase 2 Error Handling .....	328

## **Chapter 10: MODIFICATION** **337**

10.1 Parameter Modification .....	337
10.1.1 Add or Delete a System in a Unit .....	338

---

10.1.2 Add or Change Elements in a File.....	340
10.1.3 Alter Derived Element Support .....	340
10.1.4 Change Active Timespans For An Element.....	341
10.1.5 Alter the Active Timespans for a File .....	343
10.1.6 Add a New File .....	344
10.1.7 Change the Number of Work Files for DAYnnn.....	345
10.2 Standard User Exits .....	347
10.2.1 Available User Exits .....	348
10.2.2 Additional Fields Available to Exits .....	350
10.2.3 General Exits .....	352
10.2.4 Output Processing Exits.....	354
10.3 File and Data Element Customization .....	367
10.4 Testing Modifications to the Analyzer.....	368
10.5 A Case Study Using RACF and DMS/OS Data.....	368
10.5.1 Extract RACF and DMS/OS Data .....	369
10.5.2 Modify VCAGENIN .....	373
10.5.3 Modify #VCAEXIT - Define _USRUSEG Macro .....	374
10.5.4 Establish VCAACRT for User Data Elements .....	375
10.5.5 Adapting Techniques for HSM Exits .....	377
10.6 Change the Number of Work Files for DAY090 .....	379
10.7 Change the Number of Work Files for DAY095 .....	382
10.8 Enable Internal Step Restart.....	385
10.9 Implement Incremental Update.....	387

## **Appendix A: Messages** **391**

A.1 VCA Component Messages .....	391
A.2 HSM Component Messages .....	421

## **Appendix B: DATA DICTIONARY** **435**

# Chapter 1: OVERVIEW

---

The CA MICS Space Analyzer (VCA) processes detailed information about a variety of structures in MVS DASD subsystems:

- o data set information
- o catalog entry information
- o volume-level statistics
- o DFHSM migrated data sets
- o DFHSM backup versions
- o statistics on DFHSM-managed media

It organizes this information into a several files according to the data model defined by the CA MICS database.

The information gathered is essentially an inventory of the DASD subsystem at a point in time. While you can schedule this inventory as frequently as needed, most sites establish a daily inventory to feed CA MICS production databases.

The CA MICS Space Analyzer is focused on an inventory of space and total capacity without regard for the usage of data sets. That is, static data sets (those that have not been OPENed) that occupy space are measured according to the same rules as frequently accessed data sets. The focal point of this product is the occupancy of space on primary DASD or DFHSM-controlled media over a period of time.

This section contains the following topics:

[1.1 Primary Areas of Application](#) (see page 10)

[1.2 Reporting and Inquiry Facilities](#) (see page 14)

[1.3 Requirements](#) (see page 14)

## 1.1 Primary Areas of Application

The CA MICS Space Analyzer provides data integration and reporting capabilities that aid in the management of complex MVS systems in the following areas:

- o Space Management

The CA MICS Space Analyzer files provide the basis for a wide variety of analysis based on size of data set and days on non-usage. For example, some customers examine the date of last use for a data set generating HMIGRATE commands to move the data to less expensive storage based on size of data set. Smaller data sets are allowed to remain on primary storage longer than large data sets. This algorithm has been found to be more useful than the use of DFHSM daily space management in a non-SMS environment.

There are numerous other examples of analysis based on space characteristics. Identifying data sets in an excessive number of extents or VSAM data sets with an excessive number of control area splits uncovers opportunities for application-level tuning that improves both performance and (space) availability.

- o Accounting and Chargeback

The CA MICS Space Analyzer files contain data in a form that is usable for basic billing of DASD resource occupancy. A standard interface to CA MICS Accounting and Chargeback permits accounting from DASD measures of space and the special services implied by SMS class names.

- o Capacity Planning

The CA MICS Space Analyzer maintains a database of historical DASD occupancy data. This facilitates growth and trend projections at the system and organizational unit (division or department) levels.

- o Performance Management

Many of the data elements captured in this information area contain values that are important to performance management. For example, identifying volumes that do not have an indexed VTOC can offer the potential to improve performance for the entire DASD configuration. Likewise, knowing which VSAM clusters have an excessive number of

index levels provides the opportunity to boost performance of the applications that use that data.

o Operational Issues

DASD volumes tend to become fragmented with many small extents of free space over time. To make these volumes more usable, periodic jobs compacting this free space are required. Using information organized by the CA MICS Space Analyzer, you can generate control statements for this data management process only when needed and save the processing resources required for unnecessary compaction operations.

o Migration to DFSMS

Preparing for Systems Managed Storage (SMS) as implemented by DFSMS requires careful planning that the CA MICS Space Analyzer facilitates. One of the many tasks related to migrating to DFSMS requires you to identify data sets that are uncataloged or allocated with various attributes that make them unmanageable by DFSMS. You can identify these problem data sets by examining a file maintained by the CA MICS Space Analyzer and generate a control statement to either DELETE or catalog the data set.

o Management Reporting and Policy Creation

You can extract, group, and summarize information directly from the database to create useful management reporting measures. Management can establish service objectives for different types of data sets, such as storage group or volume pool, storage class, management class, or data owner. Reports or one-page management summaries that compare actual occupancy against the service level objectives create the potential for creating policies that are rational and achievable. In addition to the standard management reports that are distributed with the system, you can easily create custom reports by using online facilities. Additional tools allow resummation when the summarized data in the database does not directly address reporting needs.

Integration of Data from Catalogs, VTOCs, DFHSM and SMS

Another feature of the CA MICS Space Analyzer is its comprehensive handling of data gathered from diverse and complex structures such as all forms of MVS catalogs, VTOCs, and the SMS address space. The CA MICS Space Collector

gathers this data in a high-speed scan of these structures. Then, the CA MICS Space Analyzer:

- o Consolidates data from multiple sources providing a common access method for reporting on data set and volume occupancy. For example, a record in the VCA database which represents a data set is formed from data that originates from the VTOC, the WDS and the SMS address space. All these important attributes are organized into an easily-handled SAS observation in accordance with the CA MICS data model.
- o Interprets encoded values in the input data so that information is immediately usable in a logical form, without further conversion and/or translation.
- o Supports the classification of cost center information based on organizational structure by allowing each organization to set its own account codes to link individual data sets with the responsible cost center or organizational unit. You can use this structure to report other measured work such as batch, TSO, and printing activity with DASD and DFHSM resources by user area (engineering, payroll, manufacturing, etc.), or any other applicable section or category.
- o Eliminates duplicate data during daily processing.

### General Product Flexibility

The CA MICS Space Analyzer also offers a number of flexibility features, such as the following:

- o A wide range of standard exits to allow you to tailor the product to your specific needs.
- o A data dictionary that describes the information found in the three information areas controlled by the CA MICS Space Analyzer (VCA, HSM, and HSB).
- o Usage guidelines for applying DASD and DFHSM management information.

CA MICS database files are summarized in the same manner across all products. Therefore, you can combine batch, TSO and operational information with information about DASD occupancy and DFHSM inventories. Combining data from multiple data sources allows you to compare and correlate

batch and system information with other application-specific data such as that from RMF, CICS, IMS, DB2, etc.

The CA MICS Space Analyzer contributes to the overall benefits provided by the CA MICS IS Management Support System in the following areas:

- o Improving day-to-day IS management by allowing you to track DASD usage and plan corrective actions to ensure smooth operations
- o Reducing risk to the enterprise by providing audit and control capabilities for all DASD use
- o Controls or reduces 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
- o Improves 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
- o Increases productivity by:
  - allowing less experienced users to interpret report results and use database information
  - reducing the task of coordinating information from multiple sources so 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 Improves 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

## 1.2 Reporting and Inquiry Facilities

### Report Facilities

The CA MICS Space Analyzer provides the following reports:

- o Standard Reports provide concise information in the form of detailed or summarized tabular reports. Commands are available to filter the data to a subset of the data that would normally be presented in a given report (see Chapter 3).
- o Exception Reports allow you to define, capture, and report on exceptional conditions encountered in daily processing (see Chapter 4).
- o On-demand reporting facilities allow you to access information in the CA MICS database either interactively or in batch via the CA MICS Information Center Facility (MICF), a panel-oriented productivity tool (see Chapters 2 and 5 of this guide).

## 1.3 Requirements

The CA MICS Space Analyzer supports DFHSM data collection and processing if DFHSM Version 2.5 is installed. Version 2.6 is also supported.

# Chapter 2: USAGE GUIDELINES

---

The Space Analyzer creates and maintains a database of detailed and summary information about:

- data sets existing on primary storage
- catalog entries for VSAM data sets on primary storage
- all DASD volumes
- data sets archived to DFHSM migration levels 1 and 2
- backup versions of data sets on primary storage
- DASD volumes with daily DFHSM volume-level statistics
- tapes under the control of DFHSM

The following sections describe some usage considerations and guidelines for the data maintained by the CA MICS Space Analyzer. Some of the material here is summarized from more detailed discussions in later chapters and thus may appear to be repetitive. Our intent is to survey the "bigger picture" of usage in this chapter leaving certain details to other chapters. For example, while we present some file usage guidelines here in Chapter 2, we explain file usage in depth in Chapter 5, including diagrams of file structure and a data elements list.

This section contains the following topics:

[2.1 Methods of Using the Data](#) (see page 16)

[2.2 Management Support Applications](#) (see page 23)

[2.3 Miscellaneous Usage Notes for All Files](#) (see page 27)

## 2.1 Methods of Using the Data

There are a variety of ways you can access and use the data maintained in the Space Collector information areas. The approach you choose is usually based on several factors:

- o The questions you have about your data and the form of answer you need.
- o The availability of computed elements already in the database that can answer your question without much data analysis
- o Your experience level with SAS
- o The availability of Management Support Applications which provide additional features and reports about the data in CA MICS without any programming requirements.

In the following sections, a brief survey of the various methods of using the Space Collector information areas is presented. No one approach is best for all applications. There are times when you will want to query the CA MICS database directly by coding your request in SAS directly. Other challenges such as planning a migration to DFSMS or implementing DASD charges as part of an accounting project may be faced with greater assurance by using proven techniques and code provided by various Management Support Applications.

- 1 - Using SAS for Retrieval and Reporting
- 2 - Overview of the CA MICS Information Center Facility
- 3 - Major Features of MICF

### 2.1.1 Using SAS for Retrieval and Reporting

Perhaps the most important value in the CA MICS Space Analyzer product is the transformation of VTOC, catalog, and DFHSM data into SAS data sets. Once captured and integrated into the CA MICS database, this data is available for analysis and reporting by the extremely powerful SAS system. SAS provides a wide variety of facilities for data access and retrieval. You need not be a SAS expert to exploit the data analysis power of SAS once the data has been properly organized.

While writing your own SAS programs to access and report the data in these CA MICS information areas offers great flexibility, several facilities are provided that offer great advantages in working with SAS - even for SAS experts!

These facilities enable you to concentrate more fully on the problems to be solved while providing a variety of built-in functions and techniques that reduce what you need to know about the organization of the data. These facilities that lessen what you need to know about the data are described in the sections that follow:

- CA MICS Information Center Facility (MICF)
- CA MICS Management Support Applications

### 2.1.2 Overview of the CA MICS Information Center Facility

The CA MICS Information Center Facility (MICF) provides you with a convenient way to access and manipulate the data in the CA MICS database. Whether you are new to CA MICS or a long-term user of the database, you will find that MICF lets you rapidly obtain the results you need. With MICF, you do not need to have an in-depth knowledge of CA MICS or SAS to effectively use the CA MICS database for reporting. Using MICF avoids the problem of inadvertently coding syntax errors in SAS. In addition, MICF's automatic validation procedures keep you from attempting to analyze data elements that do not exist in the files you have requested.

The CA MICS Space Analyzer includes some standard inquiries that you will find to be immediately useful. You can readily copy these inquiries to your private MICF catalog and tailor them to your particular needs. Once you have created an inquiry, you can save and reuse it by simply selecting it from the catalog.

The optional use of color graphics provides you with the means to enhance your presentations and to assist your audience's comprehension of your results.

A comprehensive, interactive tutorial facility assists you in learning and using MICF. In addition, you have online access to all of the CA MICS documentation while you are using MICF.

When using MICF, the following major activities occur:

- o You define the input, selection criteria, and report options.
- o Then, transparent to you, MICF dynamically builds the inquiry program from your specifications.
- o Finally, the inquiry is executed either interactively or in batch, and you can display the results at your terminal or route them to a printer.

The CA MICS Space Analyzer is distributed with several fixed reports to display information from the VCA files. The reports are of a fixed format and generally show important attributes of a single file at the DETAIL or DAYS timespan.

The MICF inquiries distributed with this product serve as a data subsetting facility that can be controlled by responding to ISPF panels and table displays. MICF automatically generates all SAS code to point to the proper file and select the observations fitting the criteria you specified.

Chapter 3 of this guide describes the distributed MICF inquiries and standard reports in detail.

### 2.1.3 Major Features of MICF

MICF is a menu based system that operates under IBM's Interactive Systems Productivity Facility (ISPF). Those familiar with ISPF commands will find that MICF menus behave like ISPF menus. The major features of MICF include:

- o Structured inquiry composition
- o Logical inquiry modification
- o Direct inquiry composition
- o Execution-time specifications
- o Output replay
- o Production interface
- o Facilities for expert users
- o Distributed inquiries
- o User-written reports

Each of these features is described separately below.

#### STRUCTURED INQUIRY COMPOSITION

MICF provides a comprehensive, full-screen system that enables you to access data in the database and generate reports interactively in a structured prompt manner without requiring you to code the request in SAS.

MICF operates by automatically translating your panel specifications into an inquiry program which performs the required operations on the database. As you request data files, operations, and reports from the MICF panels, MICF performs numerous validation checks. It helps you to avoid problems by refusing to accept clearly invalid commands and by asking you to validate the accuracy of your inquiries if you have made changes to them. For example, if you add or delete steps from an inquiry, MICF will ask you to validate it before saving or executing it.

#### LOGICAL INQUIRY MODIFICATION

Once created, an inquiry can be stored in your private catalog, recalled, copied, deleted, or modified. You make changes to an inquiry using the same MICF panels as you used to create the inquiry originally. You do not, however, need to revisit panels that do not change.

#### DIRECT INQUIRY COMPOSITION

Direct inquiry composition is a feature that allows you to create an inquiry by completing a single panel. You may select from a series of pre-defined report formats and graphics templates for direct, express data retrieval and presentation.

### EXECUTION-TIME SPECIFICATIONS

You may define inquiries so that key inquiry parameters can be specified at the time an inquiry is executed. This facility allows you to report against varying data without changing the inquiry.

You can change the following parameters at execution time:

- o CA MICS database
- o CA MICS file cycles
- o Data selection criteria
- o Execution parameters
  - SAS execution parameters (whether to list SAS source code, output page dimensions, etc.)

### OUTPUT REPLAY

MICF allows you to save the output produced when you run an inquiry and replay it at another time. The output is saved by inquiry name, date, and time, so that the results of running an inquiry against different data can be saved and replayed at will. The CA MICS System Administrator can place inquiry outputs into a shared replay catalog that can be used throughout your site. You also have access to reports and color graphics from production reporting, though access to some reports may be limited by your site's security procedures.

When you replay an inquiry output, you can either view it on your terminal or send it to a hardcopy device. If you are replaying color graphics output, you do not need to use the type of device that was specified at the time the inquiry was originally executed. You can specify the color terminal, color printer, or color plotter type to use at the time you request the replay.

### PRODUCTION INTERFACE

You can generate and catalog color graphics and printed

reports with the regularly scheduled CA MICS production jobs (the daily CA MICS update). You can also control CA MICS production reporting with your site's batch scheduling facilities. MICF lets you select shared inquiries for batch execution (either with CA MICS production jobs or through your own scheduling process), with printed report and color graphics outputs being saved in a "production catalog" for later online review and printing.

Each production catalog consists of unique, user-defined printed report and color graphics data sets, so you can use your site's security facilities to limit access to sensitive reports. For example, your accounting staff may want to share a series of daily financial graphics. You can define MICF production reporting to generate and save the financial graphics during the CA MICS daily update, and you can protect the information with RACF (or another security facility).

#### FACILITIES FOR EXPERT USERS

If you are thoroughly familiar with the CA MICS database and are accustomed to programming in SAS, two features which you will find particularly helpful are Direct Inquiry Composition and the ability to include SAS code in your inquiries.

Direct inquiries are designed to allow you to compose a MICF inquiry quickly on a single panel. You are limited to one report and can only perform limited data manipulation with a direct inquiry. You will find, however, that you can complete a direct inquiry panel more quickly than you can write the corresponding SAS code. In addition, because validation functions are performed automatically by MICF and because it automatically produces error-free SAS code, you can frequently obtain the results you need with fewer coding attempts.

Facilities that include free-form SAS code as part of their MICF inquiries are available for experienced SAS programmers. You can insert any valid SAS statement at appropriate points in the SAS code that is generated by MICF. This facility gives you both the convenience of using MICF to handle routine operations and the freedom to add special coding at just the points you want it.

#### DISTRIBUTED INQUIRIES

A library of precomposed report inquiries is included with

the CA MICS Space Analyzer. You may tailor the output of these inquiries through execution time data selection.

In addition, you may develop customized reports for your site with a minimum of effort by modifying copies of the distributed inquiries. We have designed the distributed inquiry names for ease of use, each name identifies the type of output (color graphics, printer graphics, or printed reports), the component, and the timespan (daily, weekly, monthly) reported.

In this release of the CA MICS Space Analyzer, the distributed inquiries have the following characteristics:

- they are printed reports rather than color graphics
- they include the VCA information area files only
- they are designed to use the DETAIL or DAYS timespan

As stated above, you can customize MICF inquiries and thereby extend the range and utility of the CA MICS Space Analyzer reporting function. With the recent addition of support for DFHSM, we now have the opportunity of extending the reporting functions in future releases by including more distributed inquiries for VCA and HSM files.

### USER-WRITTEN REPORTS

MICF inquiry execution and output processing includes a capability to support user-written reports. With this feature, you can consolidate on demand reporting under MICF. For example, you can define a MICF inquiry that has two steps. The first step allocates the CA MICS file to be analyzed (tells MICF which file or files you want to process) and the second step refers to a file containing the SAS source statements that create your report. With this facility, you can add existing reporting routines to your MICF catalog so that the Information Systems staff only needs to go to a single location to initiate routine reporting.

## 2.2 Management Support Applications

The CA MICS Space Analyzer is classified as a Data Integration Application (DIA). The emphasis is on data reduction and integration. Recent enhancement activity has concentrated on technology support and improving usability. Relatively less emphasis is placed on data analysis.

While most clients value the flexibility and power of SAS and MICF to produce their own customized information from data contained in CA MICS, there is an increasing emphasis being placed on the "solution" and less emphasis on the "process". Thus, while SAS and MICF provide ultimate control over the "process" - the "solution" remains the responsibility of the user of the data.

To address this need for more "solution-oriented" products, we supply Management Support Applications to augment all data integration applications.

Management Support Applications are a collection of optional CA MICS products that can be used to bring a proven set of techniques and solutions to bear on problems and challenges common to all IS organizations. The Space Analyzer interfaces to:

- 1 - CA MICS Accounting and Chargeback
- 2 - CA MICS Capacity Planner
- 3 - CA MICS StorageMate

## 2.2.1 CA MICS Accounting and Chargeback

You can use CA MICS Accounting and Chargeback to apply a rate per megabyte-hour against the space occupied by a data set or group of data sets. A comprehensive series of ISPF panels guide the accounting administrator through the process of assigning rates and other setup activities.

The result of this administrative setup is a series of SAS macros that are strategically invoked by the CA MICS Space Analyzer during the DAILY run. No special pass of the data is required to compute and store the cost; accounting is, therefore, an optional but seamless extension to the processing involved in updating the CA MICS database.

The CA MICS DASD components, VCA and HSM, offer a choice of DETAIL or DAYS level accounting support. DETAIL accounting allows each individual data set to be priced according to the space occupied over time. DETAIL allows the customized accounting algorithms access to the DSNAME and VOLSER variables. This could be particularly useful to any enterprise that maintains different negotiated contracts for service levels by end-users. For example, customer x is charged the standard rate for all data sets except for all data sets beginning with "B753452.CICS.DATA\*". Standard charges for these data sets are to be discounted by 75% per a negotiated agreement. Similar deals can be created for dedicated VOLSER billing although that is not recommended with the advent of system managed storage and the widespread usage of DFHSM and other software which may move data to different volumes.

DAYS level accounting presents summarized observations to the CA MICS Accounting and Chargeback algorithms. Once summarized into a single observation according to a set of SORT/SEQUENCE variables, the DSNAME and VOLSER of a data set are lost. The aggregate of megabyte-hours for multiple data sets under a given organizational entity is presented to the accounting code.

Chapter 7 of this guide describes the parameters you must specify for a proper implementation of accounting applications. The CA MICS Accounting and Chargeback Guide discusses various methodologies that can be used with CA MICS files.

## 2.2.2 CA MICS Capacity Planner

The CA MICS Capacity Planner addresses two key areas in capacity planning: workload characterization and workload forecasting. Using data that is maintained in the CA MICS database, you are able to perform key capacity planning studies and to obtain related reports.

Based on current work activity, workload characterization helps you establish a workload baseline that will make future analyses more meaningful.

Workload forecasting helps you to estimate growth in existing applications and in resources required by new applications, and to track and revise these estimates. It uses data from a special capacity planning database that is built using data elements that you select from the CA MICS database.

In addition to providing helpful forecasting techniques, the product provides the capability for producing presentation graphics utilizing the information you obtain using these techniques. You can then develop a capacity planning strategy that is based on both past history and projections of future resource demand.

In addition to direct support for the CA MICS Space Analyzer within the CA MICS Capacity Planner, additional extensions are provided within the Capacity Planner to specifically assist in capacity planning activities within the storage subsystem. These extensions, known as Standard Applications, provide a predefined set of variables that we consider important for starting a capacity planning function in the storage arena. The Standard Applications that directly support the CA MICS Space Analyzer are:

- o The SMS Planning Standard Application
- o The DASD Planning Standard Application

### 2.2.3 CA MICS StorageMate

While the CA MICS Capacity Planner focuses on the long-term, "big picture" trends and the MONTHS timespan, CA MICS StorageMate aims at the short term problems and issues found mostly in the DETAIL and DAYS timespans. It provides over 70 reports of interest to storage administrators and their management about the status of DASD resource consumption.

Some of the reports and graphs available with this product are the following:

- o Unmanageable data set report which shows all uncataloged data sets and those with other "problem" attributes. Whether preparing for SMS implementation or performing periodic clean up on part of the DASD farm, this report enables you to pinpoint the areas of greatest payoff. Correcting job streams are automatically built for you to submit.
- o Wasted space reports which identify and quantify space wasted due to improper BLKSIZEs. Reports are factored by the amount of space allocated so that your attention is drawn to the data sets where your effort will have the greatest effect.
- o DASD/HSM data movement plot which shows the change in occupancy between active and inactive storage by application group. The daily inventory of both primary DASD and migration levels 1 and 2 is presented in a useful way which reveals to managers and end-users the total megabytes under their control across the storage hierarchy and how the distribution is changing.
- o An ACS Test Case Generator which uses a statistically valid subset of all sizes and types of data sets found in VCA and HSM information areas to build test cases for a developmental ACS routine. It shows the SMS constructs that will be assigned by the ACS code you have developed.

## 2.3 Miscellaneous Usage Notes for All Files

This section provides a convenient place to organize technical information that pertains to the product as a whole or that bridges information discussed in more detail in the CA MICS Space Collector Guide. Chapter 5 is organized to present details that are unique to each file -- here we will list items that would apply to all files maintained by the product.

Topics addressed are:

- 1 - The Inventory Nature of the Data
- 2 - The Calculation and Importance of DURATION
- 3 - Detail and Summary Data Considerations

### 2.3.1 The Inventory Nature of the Data

As you begin using the data in the various VCA and HSM database files, it is useful to understand the method by which the information is collected. The CA MICS Space Collector (VCC) literally takes an "inventory" of the DASD subsystem at a given point in time. By scanning the various control resources (VTOCs, catalogs, DFHSM control data sets) in this highly-efficient "snapshot" approach, we are able to produce a reasonably accurate inventory of data about your data.

VCC has a wide variety of recording options. When we refer to the "production" use of VCC and the "production" use of the CA MICS VCA and HSM information areas, we make the assumption that the collector has captured data about the entire DASD configuration. While it is possible and desirable to subset the DASD "farm" for special studies needed by storage administrators, these special-purpose views of the DASD farm are by their very nature, incomplete. Just as you would not call the inventory of an auto-parts store complete if you only counted the sections that containing motor oil, coolant, and floor mats, you would not deem a DASD inventory complete if you only measured a few volume pools or omitted HSM backups.

Thus, it is important to be always aware of the underlying VCC collection job stream and the options used by VCC at collection time. Try to relate one set of VCC recording options, VCC JCL (including the data sets output by VCC), and VCC scheduling frequencies with a CA MICS database unit. For example, Chapter 9 discusses some exception reports logged to the MICSLOG SYSOUT data set during DAILY processing. These clusters of messages are closely connected to the data collection function and frequently, the resolution of an exception is tied back to the data collection job output. Chapter 3 of the CA MICS Space Collector Guide describes the VCC output in great detail.

Another example of the inventory nature of the data collection relates to DFHSM. VCC as a "production inventory" should be run immediately before DFHSM "daily space management" is initiated. Because DFHSM sweeps the volumes under its management and migrates or deletes data sets according to its wide-range of rules, the inventory VCC takes will reveal day-to-day trends more clearly if it is consistently positioned relative to these DFHSM actions. Alternatively, you could schedule production VCC inventories to run immediately after the completion of DFHSM daily processing. You should avoid inconsistency.

In contrast, you can perform on-demand data collections with processing into special or test CA MICS unit databases at any time of day or at any frequency. Take care to avoid destruction of important VCC data sets used by the "production VCC".

## 2.3.2 The Calculation and Importance of DURATION

When VCC obtains an inventory of the DASD and DFHSM resources on a given day, it stores a timestamp of when data collection started and places this same timestamp in every record it produces. This timestamp becomes known to CA MICS as the data element called ENDTS.

The next time VCC runs, a new value for ENDTS is necessarily created. Each data record produced by VCC will have this current value of ENDTS and the prior value for ENDTS (this latter data element is known internally as PREV\_TS).

The difference between these two timestamps is stored in the CA MICS common element called DURATION. While DURATION is an element common to many information areas in CA MICS, its definition within the VCA, HSM and HSB information areas is simply the time between successive inventories of the data.

DURATION is a vitally important data element for these information areas because of the characteristics of space occupancy over time versus space occupancy at a given point in time. The former measurement is worthy of accounting and chargeback applications since the space occupied is being factored by how long the data set has occupied that space.

DURATION provides an efficient method of measuring space occupancy without incurring the overhead and complexity of operation associated with "event-driven" techniques such as capturing every ALLOCATE, EXTEND, PARTIAL-RELEASE, SCRATCH, and RENAME event from MVS.

DURATION calculations are maintained at the following "levels" by VCC and brought "forward" to the CA MICS Space Analyzer components VCA and HSM by including the ENDTS and PREV\_TS timestamps directly in the data records:

- 1 - at the VOLSER level
- 2 - at the DFHSM control data set level

If VCC processes a VOLSER once per day, the ENDTS and PREV\_TS values for data sets on that volume will yield a DURATION that approaches 24 hours and we would expect any variance to be negligible. Yet if, for example, a volume called DATA01 is EXCLUDED from the production VCC run for one day for some operational reason, when DATA01 is included again in the scan, the ENDTS and PREV\_TS values in the records produced for data sets on that volume will be approximately 48 hours.

### 2.3.3 Detail and Summary Data Considerations

In general, files in the DETAIL timespan will contain some identifying name such as data set name (DSNAME). As these observations are summarized, the identifying name at the detail level is discarded, numeric data elements such as tracks allocated (DAASPACA) are accumulated, and a new observation is written into the summary level timespan. The values of the sort/sequence variables determine the granularity (how many observations) at the new timespan.

Thus, at the DAYS, WEEKS, MONTHS, and YEARS timespans the CA MICS Space Analyzer will store observations with an accumulation of numeric values. You should know how to adjust these summed variables by the appropriate factors to interpret an average quantity. Let's use the HSM MIG file as an example to make this point more clear.

DETAIL Timespan

```
-----  
DSNAME          STORMGTC HSMACT1 HSMACT2 MIGALLSP MIGDSNNO  
-----  
MVS370.SP13.MACLIB STANDARD MIS-DIV SYSPROG    9.2    1  
MVS370.SP13.MODGEN STANDARD MIS-DIV SYSPROG   20.4    1  
MVS370.SP13.SMPMTS STANDARD MIS-DIV SYSPROG   10.4    1  
MVS370.SP13.SMPSTS STANDARD MIS-DIV SYSPROG   15.6    1  
MVS370.SP13.SMPPTS STANDARD MIS-DIV SYSPROG   25.8    1
```

DAYS Timespan

```
-----  
STORMGTC HSMACT1 HSMACT2 MIGALLSP MIGDSNNO  
-----  
STANDARD MIS-DIV SYSPROG    81.4    5
```

Note how DSNAME has been discarded and the commonality of the sort/sequence variables has been used to determine the observation that was created. In this highly simplistic example, our inventory has shown these 5 old data sets for an obsolete release of MVS that are stored on tape controlled by DFHSM. In the DAYS timespan, we have stored the sum of the values of MIGALLSP (allocated space in megabytes) and the number of data sets represented in the observation (MIGDSNNO).

Assuming no HRECALL or HDELETE activity is directed against these data sets, they will be detected by the next inventory

run and a new cycle in the DAYS timespan will be created with similar (if not identical) values.

Chapter 5 discusses each file structure in detail.



# Chapter 3: REPORTS

---

The CA MICS Space Analyzer provides several reporting facilities that allow you to monitor your DASD environment using either tabular reports, printer graphs, or color graphics. These reports can be executed in either batch or interactive mode, and the resulting output can be printed or displayed at your terminal.

The reports we provide are categorized as MICF standard analysis, MICF user inquiry, CA MICS management objective, and CA MICS exception reports.

The MICF standard analysis reports are a series of MICF (CA MICS Information Center Facility) inquiries provided with the CA MICS Space Analyzer. We design these reports to provide the information most commonly needed to effectively manage your DASD resources. This requires little effort on your part to produce these reports, as a large number of options are available that allow you to tailor the output to meet your needs.

MICF user inquiries are printer reports and color graphics that you compose yourself using MICF. These MICF inquiries produce meaningful reports using your CA MICS database and provide you with the flexibility of coding and saving your own reports. This facility allows you to create reports suited to your own needs with a minimum of effort.

You can produce CA MICS management reports through normal CA MICS processing and provide a concise graphic or tabular representation of your processing objectives and how well they have been met.

You can also produce CA MICS exception reports part of normal CA MICS processing. They contain an integrated and itemized list of the problems impacting your effectiveness in terms of availability, service, workload, standards, and performance. Exception reports are discussed in more detail in Chapter 4 of this guide.

The remainder of this chapter describes the operation and function of the MICF Standard Analysis reports provided with the CA MICS Space Analyzer. The other types of reports mentioned above are described in other chapters or in other CA MICS documentation.

The MICF standard analysis reports provide three levels of analysis of DASD installation data: Overview, Summary and Detail.

Overview reporting provides a short summary of key areas of Space Management. The entire DASD installation is summarized on a device-type basis. The resulting picture gives the DASD manager a concise view of who is using DASD space, how much is being used, and how effectively and efficiently it is being used.

Summary reporting provides a more detailed look at the account code and volume level. The accounts identified in the Overview Report as the largest consumers of DASD space can be listed to see who has allocated the DASD space. You can determine how the space is allocated over a set of specified volumes from the Volume Summary Report.

The last level of reporting is detail reporting. The Detail Data Set Reports for non-VSAM and VSAM data sets provide descriptive information about the data sets you specify. The information includes data set type, location, and space allocation characteristics. The Volume Track Map Report provides a detailed layout of where data set extents, free areas, and the VTOC are located on the volume. This information is useful in determining data set placement.

A new report is added to the detail reporting section that provides information about HFS file/directory of your choice. The information includes file/directory name, userid, groupid, create, access/modify timestamp, and allocation in kilobytes.

MICF standard analysis reports provide information about specific areas of DASD management. You can generate these reports from the entire database or from a specified subset of the data. This flexibility allows you to isolate particular volumes or data sets.

The reports and the sections that describe them are:

Overview report

- 1 - DASD Installation Overview Report

Summary reports

- 2 - Account Summary Report
- 3 - Volume Summary Report

Detail reports

- 4 - Detail Data Set Report
- 5 - Detail VSAM Data Set Report
- 6 - Volume Track Map Report
- 7 - HFS Detail File/Directory Report

Using reports

- 8 - Using the MICF Standard Analysis Reports

This section contains the following topics:

[3.1 DASD Installation Overview Report](#) (see page 36)

[3.2 Account Summary Report](#) (see page 40)

[3.3 Volume Summary Report](#) (see page 42)

[3.4 Detail Data Set Report](#) (see page 43)

[3.5 Detail VSAM Data Set Report](#) (see page 45)

[3.6 Volume Track Map Report](#) (see page 47)

[3.7 HFS Detail File/Directory Report](#) (see page 49)

[3.8 Using the MICF Standard Analysis Reports](#) (see page 50)

## 3.1 DASD Installation Overview Report

The DASD Installation Overview Report provides the data center manager with a one-page report showing key areas of DASD space management. The report is a composite of seven sections, each related to DASD space considerations:

- o Configuration Totals
- o Data Set Size Analysis
- o Data Set Fragmentation Analysis
- o Data Set Last Reference Analysis
- o Largest Consumers of DASD Space
- o Largest Over-Allocated Consumers of DASD Space
- o Exception Analysis

A sample report follows.

CA MICS DASD REPORTING SYSTEM									
DASD INSTALLATION OVERVIEW REPORT FOR SYSID(S) XTRA									
REPORT DATE: MONDAY, JULY 31, yyyy					DEVICE TYPE(S): 3380-K				
					REPORT TIME SPAN: 09JULyy:15:16 THRU 10JULyy:15:16				
DEVICE COUNT		DATA SET COUNT		CAPACITY (TRACKS)		...SPACE ALLOCATED... TRACKS % CAPACITY		....SPACE USED..... TRACKS % ALLOCATED	
NON-MANAGED:	0	19	0	552	.	424	.		
MANAGED:	3	975	119475	97414	81.53 %	70803	72.68 %		
TOTAL:	3	994	119475	97966	81.99 %	71227	72.70 %		
DATA SET SIZE ANALYSIS			DATA SET FRAGMENTATION ANALYSIS				DATA SET LAST REFERENCE ANALYSIS		
DATA SET COUNT	ALLOCATED TRACKS		DATA SET COUNT	ALLOCATED TRACKS	ALLOCATED EXTENTS		DATA SET COUNT	ALLOCATED TRACKS	LAST REFER. (MONTHS)
+-(% TOTAL)--			+-(% CAPACITY)+				+-(% TOTAL)-		+-(% CAPACITY)+
6 (1%)	2000 & OVER		5 (1%)	5250 (4%)	13-16		166 (17%)	6617 (6%)	13 & OVER
14 (1%)	1000-1999		5 (1%)	4081 (3%)	9-12		9 (1%)	356 (0%)	10-12
25 (3%)	500- 999		24 (2%)	20131 (17%)	5- 8		7 (1%)	734 (1%)	7- 9
52 (5%)	250- 499		91 (9%)	28780 (24%)	2- 4		15 (2%)	610 (1%)	4- 6
51 (5%)	100- 249						101 (10%)	9892 (8%)	1- 3
148 (15%)	_TOTALS_		125 (13%)	58242 (49%)	_TOTALS_		298 (30%)	18209 (15%)	_TOTALS_
LARGEST CONSUMERS OF DASD SPACE					LARGEST OVER-ALLOCATED CONSUMERS OF DASD SPACE				
DIVISION	DATA SET COUNT	ALLOCATED TRACKS			DIVISION	ALLOCATED TRACKS	UNUSED TRACKS		
+-(% TOTAL)-		+-(% CAPACITY)+			+-(% CAPACITY)+		+-(% ALLOCATED)+		
TSE	416 (42%)	47170 (39%)			ICC	836 (1%)	672 (80%)		
CDI	37 (4%)	8522 (7%)			DTB	2047 (2%)	1967 (96%)		
BDV	64 (6%)	4360 (4%)			UPF	4075 (3%)	1979 (49%)		
UPF	23 (2%)	4075 (3%)			CDI	8522 (7%)	4278 (50%)		
DTB	7 (1%)	2047 (2%)			TSE	47170 (39%)	5459 (12%)		
_TOTALS_	547 (55%)	66174 (55%)			_TOTALS_	62650 (52%)	14355 (23%)		
EXCEPTION ANALYSIS									
(09001) NO EXCEPTIONS									
(09002) NO EXCEPTIONS									
(09003) NO EXCEPTIONS									
(09004) NO EXCEPTIONS									
(09005) 5 DATA SET(S) WHERE DATA SET HAS TOO MANY EXTENTS. EXTENT COUNT= 14									

Figure 3-1. DASD Installation Overview Report

#### Configuration Totals Section

This section gives configuration totals for the number of data sets allocated. It also provides totals for the number of DASD tracks in the system (capacity), tracks allocated, and tracks in use. Data set and volume counts are also reported based on whether the volume or data set is being managed by System Managed Storage (SMS).

#### Data Set Size Analysis Section

This section groups data sets into one of five groups according to data set size. For each group, totals are produced for the count of data sets in the group. The data set count is also expressed as a percentage of the total data set count, as reported in the Configuration Totals section.

The data in this section is useful in determining the number of data sets that have a large amount of space allocated.

#### Data Set Fragmentation Analysis Section

This section groups data sets according to the number of extents they have allocated. The number of extents allocated is divided into three groups. For each group, totals are produced for the count of data sets and the amount of space allocated. The space allocated is also listed as a percentage of the total DASD space as reported in the Configuration Totals section.

The data in this section shows how many data sets are inefficiently allocated.

#### Data Set Last Reference Analysis Section

This section groups data sets into one of five groups according to the last date referenced. For each group, totals are produced for the count of data sets and the amount of space allocated. The space allocated is also listed as a percentage of the total DASD space, as reported in the Configuration Totals section.

The data in this section is useful in determining how much DASD space is allocated to infrequently used data sets.

#### Largest Consumers of DASD Space Section

This section of the report lists the high-level account code for the five largest users of DASD space. The data is ordered according to the amount of DASD space allocated. For each account code, totals are produced for data set count and space allocated. The space allocated is also listed as a percentage of the total DASD space as reported in the Configuration Totals section. The last line of the section contains totals for the five accounts listed.

The data in this section shows which accounts have the largest impact on DASD space allocation.

#### Largest Over-Allocated Consumers of DASD Space Section

This section lists the high-level account code for the five users of DASD space who have the most allocated space not in use. The data is ordered according to the percentage of allocated space not in use. For each account code, totals are produced for space allocated and space used. The space allocated is also listed as a percentage of the total DASD space as reported in the Configuration Totals section. Additionally, the space used is listed as a percentage of the space allocated. The last line of the section contains totals for the five accounts listed.

This data is useful in determining which accounts have the largest amounts of space not in use.

#### Exception Analysis Section

This section lists the status of the five most critical exceptions that have been defined in the VCA information area. A count of the number of volumes that have generated each exception is also produced.

The data in this section is useful in obtaining the status of the five areas you think are most critical for your site.

#### Tailoring Options

See Section 3.8.3 of this guide to see how to customize the information on the report to better meet your needs.

## 3.2 Account Summary Report

The Account Summary Report provides a listing of DASD space allocation by account codes. Multiple levels of account codes can be processed. The report sequence is account code level one, account code level two, then device type. For each level of account code, totals are produced for data set count, space allocated, and space used. The space used is also listed as a percentage of the space allocated. The listing is subdivided by device type, with a line for each device type on which the account has allocated data sets.

For each high-level account code, a subtotal is produced for all of the lower levels of account codes listed. A final total is produced for all high-level account codes listed.

Use this data to determine the percentage of the DASD resource used by all account codes.

A sample report follows.

CA MICS DASD REPORTING SYSTEM				
ACCOUNT SUMMARY REPORT FOR SYSID(S) XTRA				
REPORT DATE:	MONDAY, JULY 31, yyyy	REPORT TIME SPAN: 09JULyy THRU 10JULyy		
DIVISION	DEVICE	DATA SET	ALLOCATED	USED
PROJECT	TYPE	COUNT	TRACKS	TRACKS
				(% ALLOCATED)
TSE				
AIC	3380-K	12	672	600 (89%)
ALC	3380-K	22	944	794 (84%)
AMC	3380-K	1	1	1(100%)
BDB	3380-K	35	14766	14703(100%)
BCD	3380-K	4	40	40(100%)
BD	3380-K	57	597	590 (99%)
BKD	3380-K	2	382	382(100%)
BLD	3380-K	2	135	116 (86%)
CDE	3380-K	1	5	0 (0%)
CIF	3380-K	4	677	658 (97%)
CRF	3380-K	5	50	50(100%)
DLG	3380-K	4	3526	3513(100%)
EBI	3380-K	3	183	183(100%)
EOI	3380-K	14	1633	1258 (77%)
FAL	3380-K	1	153	143 (93%)
FFM	3380-K	11	1301	877 (67%)
FMS	3380-K	2	53	53(100%)
GEN	3380-K	3	65	56 (86%)
HAP	3380-K	20	5710	5554 (97%)
HEP	3380-K	1	112	105 (94%)
HMP	3380-K	2	405	388 (96%)
HUP	3380-K	3	288	276 (96%)
IMS	3380-K	1	20	20(100%)
IS	3380-K	10	657	638 (97%)
ITS	3380-K	9	1459	1401 (96%)
JDT	3380-K	41	3808	3276 (86%)
_SUBTOTALS_		416	47170	41711 (88%)
TSM				
***	3380-K	2	16	1 (6%)
_SUBTOTALS_		2	16	1 (6%)
_TOTALS_		994	97966	71227 (73%)

Figure 3-2. Account Summary Report

### 3.3 Volume Summary Report

The Volume Summary Report provides information related to the status of free space on DASD volumes and the status of the volumes' VTOCs. The report sequence is device type, then volume serial. For each volume you select, a detail line is produced that contains totals for free space, count of free extents, number of tracks unaccounted for (missing), number of alternate tracks in use, and also a fragmentation index.

A sample report follows.

CA MICS DASD REPORTING SYSTEM													
VOLUME SUMMARY REPORT FOR SYSID(S) XTRA													
REPORT DATE: MONDAY, JULY 31, yyyy				REPORT TIME SPAN: 09JULyy:15:16 THRU 10JULyy:15:16									
S		SPACE						VTOC					
VOLUME	DEVICE	CAPACITY	TOTAL FREE	LARGEST FREE	FREE EXT	LOST TRK	USED	FRAGMENT	TOTAL	USED	FREE	SPREAD	
*SERIAL	TYPE	(TRACKS)	(TRACKS)	(TRACKS)	COUNT	COUNT	ALT TRK	INDEX	DSCB	DSCBS	DSCBS	INDEX	
			(% CAPACITY)	(% CAPACITY)					(% TOTAL)	(% TOTAL)			
C SMS001	3380-K	39825	4118 (10%)	230 (1%)	187	0	0	0.609	3180	324 (10%)	2856 (90%)	1.0	
C SMS002	3380-K	39825	12778 (32%)	3015 (8%)	150	0	0	0.385	3127	286 (9%)	2841 (91%)	1.0	
C SMS003	3380-K	39825	4432 (11%)	316 (1%)	183	0	0	0.555	3127	442 (14%)	2685 (86%)	1.0	
VOLUMES		3											
_TOTALS_		119475	21328 (18%)	3561 (3%)	520	0	0		9434	1052 (11%)	8382 (89%)		

Figure 3-3. Volume Summary Report

An asterisk (\*) in the first report column indicates that the DIRF bit is set in the VTOC for that volume. Refer to Exception 09013 in Chapter 4 for an explanation of this indication. Other possible values for this column include a 'C' (meaning the volume has been converted to full SMS management), or an 'I' (meaning that conversion to SMS management is in process).

The fragmentation index gives an indication of how fragmented a volume has become and whether or not it is a candidate for compaction. The higher the fragmentation value, the more a volume needs to be compacted. Also, for each volume, totals are produced for number of Data Set Control Blocks (DSCBs) in the VTOC, number of DSCBs in use, and number of available DSCBs.

A "spread" index is also produced, giving a numeric indication of how spread out the used DSCBs are in the VTOC. Longer search times will result when a VTOC's used DSCBs are not concentrated at the front of the VTOC. A line showing totals is also produced.

Use this report for obtaining a concise look at the state of DASD volumes and how efficiently they can meet future allocation demands.

## 3.4 Detail Data Set Report

The Detail Data Set Report provides detailed information concerning selected data sets. The report sequence is volume serial, then data set name. For each data set, a detail line is produced that contains volume information, DCB information, creation and last use dates, and allocation information. The allocation information contains space allocated and space used. The space used is also produced as a percentage of space allocated. A total for all listed data sets is also produced.

Use this report to determine the type and allocation characteristics for a specific data set.

The figure below is an example of this report.

3.4 Detail Data Set Report

CA MICS DASD REPORTING SYSTEM														
DETAIL DATA SET REPORT FOR SYSID(S) XTRA														
REPORT DATE:		MONDAY, JULY 31, yyyy						REPORT TIME SPAN: 09JULyy:15:16 THRU 10JULyy:15:16						
S	M	DATA SET NAME	VOLUME	DEVICE				CREATE	LASTUSE	ALLOC	NO	ALLOCATED	USED	
S			SERIAL	TYPE	DSORG	RECFM	LRCL	BLKSIZE	DATE	DATE	TYPE	EX	TRACKS	TRACKS
													(% ALLOCATED)	
		*QRG9060.SPFL0G1.LIST	SMS002	3380-K	PS	VA	125	129	10JULyy	10JULyy	BLK	1	8	0 (0%)
		*QRP8318.PRINT.KEEP	SMS002	3380-K	PS	FBA	121	121	03JULyy	10JULyy	TRK	2	9	8 (89%)
		*QRS.COENDJOB.SRCM0S2F.BJS1PERM	SMS002	3380-K	PS	FB	80	23440	03NOVyy	05JULyy	TRK	1	1	1 (100%)
		*QRS.CSENDJOB.VARM0S3F.BJS1PERM	SMS002	3380-K	PS	FB	80	23440	03NOVyy	16JUNyy	TRK	1	2	2 (100%)
		*QRS.INENDJOB.SVAM0S4F.BJS1PERM	SMS002	3380-K	PS	FB	80	23440	03NOVyy	07JULyy	TRK	1	3	3 (100%)
		*QRS.INFOCNTR.ACTRIS0F.BJS1PERM	SMS002	3380-K	PS	FBA	133	15428	30JUNyy	06JULyy	BLK	1	2	2 (100%)
		*QRS.NEENDJOB.SVAM0S4F.BJS1PERM	SMS002	3380-K	PS	FB	80	23440	03NOVyy	16JUNyy	TRK	1	3	3 (100%)
		*QRS.TIENDJOB.MRGM1S0F.BJS1PERM	SMS002	3380-K	PS	FB	1020	23460	06FEByy	07JUNyy	TRK	6	1255	1255 (100%)
		*QSB2309.PRD.DSHEX640	SMS002	3380-K	PS	FB	130	15470	10JULyy	10JULyy	TRK	1	5	5 (100%)
		*QSB2309.TEST.XPAMASTR	SMS002	3380-K	PS	VB	8192	23476	10JULyy	10JULyy	TRK	4	60	60 (100%)
		*QSL6018.SALINC.FILESIN	SMS002	3380-K	PS	FB	116	1160	05JULyy	07JULyy	TRK	1	1	1 (100%)
		*QTA8015.SPFL0G1.LIST	SMS002	3380-K	PS	VA	125	129	10JULyy	10JULyy	BLK	1	8	1 (13%)
		*QTH2757.FROGPGC.JUN20yy.DBEXT001	SMS002	3380-K	PS	FB	624	23088	10JULyy	10JULyy	CYL	2	690	676 (98%)
		*QTM4953.SPFL0G1.LIST	SMS002	3380-K	PS	VA	125	129	10JULyy	10JULyy	BLK	1	8	0 (0%)
		*QUF.ACTIVITY.RPTR0S0F.BJS1GDG. G0446V00	SMS002	3380-K	PS	FBA	133	133	10JULyy	10JULyy	TRK	1	3	3 (100%)
		*QUF.DLYLG002.PFFF0S0F.BJS1GDG. G1259V00	SMS002	3380-K	PS	FB	168	15456	04JULyy	04JULyy	TRK	1	1	1 (100%)
		*QUF.LNK.TUTORIAL	SMS002	3380-K	PS	FB	80	19040	19MAYyy	16JUNyy	TRK	1	1	1 (100%)
		*QUF.MAINTOPS.PFFI0S0F.BJS1GDG. G0064V00	SMS002	3380-K	PS	FB	245	245	05JULyy	05JULyy	TRK	1	10	1 (10%)
		*QWC5842.SPFL0G1.LIST	SMS002	3380-K	PS	VA	125	129	10JULyy	10JULyy	BLK	1	8	0 (0%)
		*QWR5571.DORD.PACWW.SCREEN	SMS002	3380-K	PS	FBA	81	810	10JULyy	.	TRK	1	5	0 (0%)
		*QWW9982.SPFL0G1.LIST	SMS002	3380-K	PS	VA	125	129	10JULyy	10JULyy	BLK	1	8	1 (13%)
		*SMF.BKMHICIS.SMFI0S0F.BJS1GDG. G0255V00	SMS002	3380-K	PS	VBS	32767	4096	10JULyy	10JULyy	CYL	1	210	204 (97%)
		*SSU.UCC7WLB4.TMSR0S0I.BJS1PERM	SMS002	3380-K	PS	FBA	133	3325	02MAYyy	10JULyy	CYL	3	210	206 (98%)
		-SYS1.VT0CIX.VSMS002	SMS002	3380-K	PS	F	2048	2048	31MARyy	.	TRK	1	14	14 (100%)
		-SYS89191.T145820.RA000.PMT2622X. PAYPS174	SMS002	3380-K	PS	FB	290	6380	10JULyy	10JULyy	CYL	1	15	8 (53%)
		-SYS89191.T151443.RA000.PDJ6438D. R0000000	SMS002	3380-K	PS	U	0	0	10JULyy	10JULyy	CYL	1	300	45 (15%)
		*TESTPCH.BACKUP00.PCHE9066.BJS1GDG. G0111V00	SMS002	3380-K	PS	VB	23472	23476	08JULyy	08JULyy	CYL	1	600	599 (100%)
		*TESTPCH.BACKUP00.PCHE9068.BJS1GDG. G0112V00	SMS002	3380-K	PS	VB	23472	23476	10JULyy	10JULyy	CYL	4	105	99 (94%)
		*TESTPCH.BACKUP00.PCHE9099.BJS1GDG. G0111V00	SMS002	3380-K	PS	VB	23472	23476	08JULyy	08JULyy	TRK	1	1	1 (100%)
		*TESTSPM.GLTEST.REPORTS	SMS002	3380-K	PS	FBA	133	23408	18MAYyy	10JULyy	CYL	1	15	6 (40%)
		*TESTSPM.RSTGL1D2.SPMM0S0I.BJS1DATA. G0027V00	SMS002	3380-K	PS	FB	180	23400	08JULyy	08JULyy	CYL	1	15	6 (40%)

Figure 3-4. Detail Data Set Report

## 3.5 Detail VSAM Data Set Report

The Detail VSAM Data Set Report provides detailed information concerning selected data sets. The report sequence is volume serial, then data set name. For each data set, a detail line is produced that contains information pertaining to specific VSAM allocation considerations. The data set organization, either entry-sequenced data set (ESDS) or key-sequenced data set (KSDS), is listed, as is the information concerning the configuration of the control intervals and control areas.

Use this report to monitor VSAM data sets as they change in size and reorganize data sets if space becomes poorly utilized.

3.5 Detail VSAM Data Set Report

CA MICS DASD REPORTING SYSTEM															
DETAIL VSAM DATA SET REPORT FOR SYSID(S) XTRA															
REPORT DATE:		MONDAY, JULY 31, yyyy				REPORT TIME SPAN: 09JULy:15:16 THRU 10JULy:15:16									
S	M	S													
	DATA SET NAME	VOLUME SERIAL	DEVICE TYPE	RECFM	AVG. LRECL	BUFSIZE	PER CA	TOTAL CI'S	% CI'S USED	CI	CI'S PER CA	FREE CI/CA	FREE BYTES IN CI	CI	CA
	*RUN32MFS.ONLINE.PEARR008.DATA	SMS001	3380-K	KSDS	48	9728	15	750	(20%)	4096	150	7	409	0	0
	*RUN32MFS.ONLINE.PEARR008. INDEX	SMS001	3380-K	KSDS	0	0	1	23	(4%)	1536	23	0	0	0	0
	*RUN32MFS.ONLINE.XXCRT012.DATA	SMS001	3380-K	KSDS	264	9728	15	4500	(20%)	4096	150	7	409	0	0
	*RUN32MFS.ONLINE.XXCRT012. INDEX	SMS001	3380-K	KSDS	0	0	1	46	(15%)	1536	23	0	0	0	0
	*RUN32PAY.ONLINE.IEC91988. IEFMCMF.DATA	SMS001	3380-K	KSDS	80	8704	1	10	(0%)	4096	10	0	0	0	0
	*RUN32PAY.ONLINE.IEC91988. IEFMCMF.INDEX	SMS001	3380-K	KSDS	0	0	1	46	(0%)	512	46	0	0	0	0
	*RUN32PMF.ONLINE.NFTAMVTM. MINM0D0G	SMS001	3380-K	KSDS	1600	17408	14	70	(100%)	8192	70	10	1228	0	0
	*RUN32PMF.ONLINE.NFTAMVTM. MINM0I0G	SMS001	3380-K	KSDS	0	0	1	1	(0%)	1024	1	0	0	0	0
	RUN32PMF.ONLINE.NFTAMVTM. MINM0I0G	SMS001	3380-K	SSET	0	0	1	1	(100%)	1024	1	0	0	0	0
	*RUN32TDE.ONLINE.DATA.QUBRTI0G	SMS001	3380-K	KSDS	191	6656	15	8100	(57%)	2048	270	13	614	0	0
	*RUN32TDE.ONLINE.INDEX. QUBRTI0G	SMS001	3380-K	KSDS	0	0	1	45	(40%)	2560	15	0	0	0	0
	*RUN32TOP.ONLINE.VSMDATA. TOPAS01X	SMS001	3380-K	KSDS	149	10240	3	390	(15%)	4096	30	1	0	0	0
	*RUN32TOP.ONLINE.VSMINDX. TOPAS01X	SMS001	3380-K	KSDS	0	0	1	90	(1%)	2048	18	0	0	0	0
	RUN32TOP.ONLINE.VSMINDX. TOPAS01X	SMS001	3380-K	SSET	0	0	1	13	(89%)	2048	1	0	0	0	0
	*KIC.CICS.DATA.DFHTEMP	SMS002	3380-K	ESDS	4089	8192	15	300	(100%)	4096	150	0	0	0	0
	*KIC.ONLINE.REGION3.DFHCS.D DATA.PJK2PERM	SMS002	3380-K	KSDS	100	45568	9	18	(100%)	22528	18	0	0	2	0
	*KIC.ONLINE.REGION3.DFHCS.D INDEX.PJK2PERM	SMS002	3380-K	KSDS	0	0	1	46	(2%)	512	46	0	0	0	0
	SYS1.VVDS.VSMS002	SMS002	3380-K	ESDS	0	0	1	300	(100%)	4096	10	0	0	0	0
	*TESTDBA.DBIS.MIG21.VSAMEXT.D	SMS002	3380-K	KSDS	164	17408	15	300	(0%)	8192	75	0	0	0	0
	*TESTDBA.DBIS.MIG21.VSAMEXT.I	SMS002	3380-K	KSDS	0	0	1	31	(0%)	1024	31	0	0	0	0
	*TESTWSE.ONLINE.KEYPLAD.V2R1M0. FORMAT.D	SMS002	3380-K	RRDS	4000	20480	15	4050	(100%)	4096	150	0	0	0	0
	*TESTWSE.ONLINE.KEYPLAD.V2R1M0. KEYED.D	SMS002	3380-K	RRDS	3000	15360	15	1170	(100%)	3072	195	0	0	0	0
	*VLOX.AS.DISTATRN.DATA	SMS002	3380-K	ESDS	4089	8192	15	9600	(100%)	4096	150	0	0	0	0
	*VLOX.DSNLOX.NBCDT00T.NBCTS132. I0001.A001	SMS002	3380-K	LDS	0	8192	2	120	(83%)	4096	20	0	0	0	0
	*VLOX.DSNLOX.DSNDB23.TEMPH5. I0001.A001	SMS002	3380-K	ESDS	4089	8192	1	80	(100%)	4096	10	0	0	0	0

Figure 3-5. Detail VSAM Data Set Report

## 3.6 Volume Track Map Report

The Volume Track Map Report provides a complete description of how space is being utilized on a DASD volume. A line is produced for each allocated or free extent on the volume. The extents are ordered by track address. If the extent is allocated, the detail line will include the data set name, the extent number of this extent and the total number of extents allocated to the data set, the track start address of the extent, and the size of the extent in tracks. Free extents are also listed, as is the VTOC.

Use this report to determine the location and size of free space extents. Also use it to determine how well data sets are allocated with respect to cylinder alignment and numbers of extents, and where clusters of data sets are located with respect to the beginning of the volume or to the VTOC.

This report may not produce the desired results when used with the MICF Production Reporting facility. This is because, by default, the report will only select volumes with a serial number prefix of "TS0" for reporting. In most cases this is not a problem, because this report is normally used for specific problem research, and not as part of regular production reporting. If you want to use this report in a production reporting environment, you may have to copy the inquiry, edit the default value to meet your standards, and specify the modified inquiry as input to the MICF Production Reporting facility.

3.6 Volume Track Map Report

CA MICS DASD REPORTING SYSTEM									
TRACK MAP REPORT FOR 3380-K DEVICE, VOLUME SMS002									
REPORT DATE: MONDAY, JULY 31, yyyy					REPORT TIMESPAN: 09JULyy:15:16 THRU 10JULyy:15:16				
S		EXTENT	TRACK	TRACK	S		EXTENT	TRACK	TRACK
M	DATA SET NAME	NUMBER	START	COUNT	M	DATA SET NAME	NUMBER	START	COUNT
S					S				
	*** LABEL ***		0	1		*JOBMLGM.GLMPI021.REPORT	3/16	167	1
	----- V T O C -----		1	59		*PMS.CP000004.PMSM051F.WXT1GDG.G3935V00	1/ 1	168	6
	-SYS1.VT0CIX.VSMS002	1/ 1	60	14		*TESTCFS.CHDBS.CFSR533	1/ 1	174	5
	*SYS1.VVDS.VSMS002	1/ 1	74	30		*PFU.LNK.TUTORIAL	1/ 1	179	1
	*RUN32TOP.ONLINE.VALUEIDX.TOPSPINV	1/ 1	104	1		*JOBMLGM.GLMPI021.REPORT	4/16	180	1
	*RPQ1330.LQE	1/ 1	105	1		*PMF7584.OLRJRTRN.INDEX	1/ 1	181	1
	*TESTFCS.ONLINE.FCSRCCPR	1/ 1	106	1		*JOBMLGM.GLMPI021.REPORT	5/16	182	1
	*TESTDAT.TST.DATEN2W6.RPT@1576.G0163V00	1/ 1	107	2		*RUN32CMF.ONLINE.MATRX999.MFDM0D0G	1/ 1	183	2
	*DMV.VWXTLB30.DMSB050I.WXT1DATA.G0657V00	1/ 1	109	1		*PPG3403.TRANS030.CDFE2S0I.WXT1GDG.G0003V00	1/ 2	185	1
	*TESTPCH.BACKUP00.PCHE9099.WXT1GDG.G0111V00	1/ 1	110	1		*PMH6825.DSNORUN.CNTL	1/ 1	186	2
	*RUN320CS.ONLINE.NEW.OCS0008.INDEX	1/ 1	111	1		*PMS.ALIASPNT.PMSE050I.WXT1DATA.G1943V00	1/ 1	188	1
	*TESTFCS.DACBS.SRTPR447	1/ 1	112	1		*JOBMLGM.GLMPI021.REPORT	6/16	189	1
	*RUN32PAY.ONLINE.IEC91988.IEFMRTE.DATA	1/ 1	113	1		*ICI.RMDS7774.ICIR050G.WXT1PERM	3/ 5	190	5
	*TESTFCS.DACBS.CFSR449	1/ 3	114	1		*PMS.CP000009.PMSM051F.WXT1GDG.G3798V00	1/ 1	195	11
	*JDBT.DSN0BD.WHTISTBD.ACCTEX03.I0001.A001	1/ 1	115	1		*PFU.MAINTOPP.PFUI050F.WXT1GDG.G0064V00	1/ 1	206	10
	*RUN32PAY.ONLINE.IEC91988.IEFMRTE.INDEX	1/ 1	116	1		*PJD8003.PRBLNFLT	1/ 1	216	5
	*DCI.PMICS.MAMAIN.USER.LOAD	1/ 1	117	2		*PMS.VCICSD10.PMSB050F.WXT1GDG.G0340V00	1/ 1	221	2
	*QFU.ACTIVITY.RPTRS05F.WXT1GDG.G0446V00	1/ 1	119	3		*PMS.CP000009.PMSM051F.WXT1GDG.G3800V00	1/ 1	223	11
	*TESTCFS.CADBS.CFSR413	1/ 1	122	3		*EFGI.RT.UZD.DDATN2W6.RPT#5174.G0161V00	1/ 1	234	1
	*TESTCLN.USERTST.DBM.FDBSTATS	1/ 1	125	1		*JOBMLGM.GLMPI021.REPORT	7/16	235	1
	*TESTCFS.CADBS.SRTPR449	1/ 3	126	1		*JOBMLGM.GLMPI021.REPORT	8/16	236	1
	*QMN6332.WK.SUPPORT.JUNyy	1/ 1	127	1		*EFGI.RT.UZD.RCPST300.DA2E0S2F.G0153V00	1/ 1	237	1
	*RUN32TDS.VEESOL.SYSOPT.INDEX	1/ 1	128	1		*EFGI.RT.UZD.DDATN1W6.RPT#3074.G0169V00	2/ 2	238	1
	*QMF7458.PRODDATE.FILE	1/ 1	129	1		*TESTCFS.CHDBS.RPT@R956	1/ 1	239	5
	*** FREE SPACE ***		130	1		*JOBMLGM.GLMPI021.REPORT	9/16	244	1
	*QRS.INFOCNTR.BASRTS0F.WXT1PERM	1/ 1	131	2		*PMH6825.NBC.DBNLYZR.FILELDEL	1/ 1	245	1
	*TESTCFS.CFSPD556.CNTRYCD	1/ 1	133	1		*ICI.RMDS7774.ICIR050G.WXT1PERM	1/ 5	246	1
	*QRP8183.PRINT.KEEP	2/ 2	134	1		*EFGI.RT.UZD.CRTCK090.DA2E0S0I.G0260V00	1/ 3	247	5
	*DCI.MICSP.VMAIN3.CHECKPT.DATA	1/ 1	135	2		*EFGI.RT.UZD.CRTCK090.DA2E0S0I.G0260V00	2/ 3	252	3
	*JOBMLGM.GLMPI021.REPORT	1/16	137	1		*EFGI.RT.UZD.CRTCK090.DA2E0S0I.G0260V00	3/ 3	255	1
	*QJD1358.PN431V.COBOL	1/ 1	138	1		*JOBMLGM.GLMPI021.REPORT	10/16	256	1
	*RUN32TDE.ONLINE.ALTXIDX.PERADI0G	1/ 2	139	1		*JOBMLGM.GLMPI021.REPORT	11/16	257	1
	*RUN32TDE.ONLINE.ALTXIDX.PERADI0G	2/ 2	140	1		*RUN32DDA.JLTRNSTB.INDEX	1/ 1	258	3
	*PMS.VTMSD810.PMSB050F.WXT1GDG.G0811V00	1/ 1	141	1		*JOBMLGM.GLMPI021.REPORT	12/16	261	1
	*QLH3551.NEW.KASCO	1/ 1	142	2		*PRP8183.PRINT.KEEP	1/ 2	262	8
	*TESTTDS.RRS891.CICTX060	1/ 1	144	5		*TESTSST.REQUEST.\$01852	1/ 1	270	2
	*STS.VIAINS.R32.VIACEN12.PTF	1/ 1	149	3		*JOBMLGM.GLMPI021.REPORT	13/16	272	1
	*TESTFRP.EDI80004.RESTONVA	1/ 1	152	1		*ICI.RMDS7774.ICIR050G.WXT1PERM	2/ 5	273	5
	*RUN32CMF.ONLINE.MATIX696.MFDM0I0G	1/ 1	153	1		*PMS.CP000003.PMSM051F.WXT1GDG.G4103V00	1/ 1	278	16
	*JOBMLGM.GLMPI021.REPORT	2/16	154	1		*TESTCFS.CNDBS.CFSPU425	3/ 3	294	5
	*RUN32LAS.ONLINE.LSOSERR.DATA	1/ 1	155	5		*JOBMLGM.GLMPI021.REPORT	14/16	299	1
	*EFGI.RT.UZD.DDATN1W6.RPT#3074.G0169V00	1/ 2	160	1		*PLH3551.OLD.ADVRRATES	1/ 1	300	4
	*EFGI.RT.UZD.DDATN2W6.RPT#5170.G0163V00	1/ 1	161	1		*TESTFCS.CNDBS.RPTPR603	1/ 1	304	5
	*QPG3403.TRANS010.CDFE2S0I.WXT1GDG.G0003V00	2/ 2	162	5		*JOBMLGM.GLMPI021.REPORT	15/16	309	1

Figure 3-6. Volume Track Map Report



## 3.8 Using the MICF Standard Analysis Reports

This section describes the procedures you should use to execute the MICF standard analysis inquiries, and tailor them for your specific environment. The following sections include the following topics:

- 1 - Executing the MICF Standard Analysis Inquiries
- 2 - Execution Time Override Options
- 3 - Macro Variable Override Options
- 4 - Sample Inquiry Execution
- 5 - MICF Production Reporting Support

### 3.8.1 Executing the MICF Standard Analysis Inquiries

The MICF Standard Analysis Reports are accessed through the Database Inquiries option of MICF, which is part of the CA MICS Workstation Facility (MWF). Contact your CA MICS Administrator if you are unsure how to access MWF through your ISPF system.

If you have not used MICF before, you should select the MICF Options feature of MICF. This allows you to set options that affect the execution and presentation of your inquiries. This allows you to set environmental options, such as whether reports should be run in background or interactive mode.

The report inquiries are stored in your Shared MICF catalog in a Catalog Group named STORAGE. The names of the inquiries, the reports they produce, and the files and timespans used by each inquiry are shown in Figure 3-7 below:

Inquiry	Report Name Produced	File(s) Used
VCALD1	DASD Installation Overview Report	DETAIL.VCADAA DETAIL.VCA_VS DETAIL.VCAVOA DAYS.ADMEXC
VCALD2	Account Summary Report	DAYS.VCADAA DAYS.VCA_VS
VCALD3	Volume Summary Report	DETAIL.VCAVOA
VCALD4	Detail Data Set Report	DETAIL.VCADAA
VCALD5	Detail VSAM Data Set Report	DETAIL.VCA_VS
VCALD6	Volume Track Map Report	DETAIL.VCADAA DETAIL.VCAVOA

Figure 3-7. Report Inquiry Descriptions

### 3.8.2 Execution Time Override Options

After you have selected an inquiry for execution, you will be presented with one or more screens of execution time override options. While most of these values are optional, or will use default values, they can also be used to tailor the data that will appear on the final reports. When multiple options are specified, they must all be satisfied before data is selected. For example, if you requested data sets with a DFSMS Storage Class of NORMAL and an organization (DSORG) of PS, only those data sets having BOTH a Storage Class of NORMAL and a DSORG of PS would appear in the resulting report. This section will describe each of the override options that you might see when you execute an inquiry. The function and syntax of each option will be described, and examples will be shown.

The general format of each override option is shown in the example below:

```
Select by Data Set Name:      EQ/NE - value* ...  
EQ                           -----
```

The first line describes the function of the option and the syntax of the two operands on the second line. The second line contains two operand areas that you will supply if you want to use this option. In this example, the first line indicates this option can be used to select only certain data sets to appear on this report. The first line also describes the syntax of the two operands that can be coded on the second line. In this example, the first operand (the one that appears in the left column), must have a value of either EQ (EQUAL) or NE (NOT EQUAL). The default value of EQ already appears in the operand location. This represents the type of comparison that will be done between the data set name and the value in the second operand. For example, if you code EQ as the first operand, the data set name must be EQUAL to the value specified in the second operand, or it will not be selected. The second operand specifies the value the data set name will be compared against. The three periods (...) indicate that multiple values can be present. The asterisk (\*) indicates that a short, or prefixed value can be entered by specifying an asterisk as the last character. Assume you wanted to include all data sets in your report except a data set named SECRET and all data sets with the prefix PAY. You could accomplish this by coding the option statement as shown below:

Select by Data Set Name: EQ/NE - value\* ...  
NE SECRET PAY\*

#### Line Size and Page Size Options

-----

MICF provides options to allow the specification of default output line size and page size. As all of the standard reports use up through report column 132, any MICF default line size you establish will be ignored (a value of 132 will be forced for these reports only).

MICF defaults for page size (number of lines per page) will be honored. However, if you choose a page size less than 60, make sure the value you specify is large enough to account for the heading and title lines that appear on each page.

Each of the possible execution time override options will now be described. Note that some options will apply to all inquiries, while others apply to one or two specific inquiries. These options are listed in alphabetical order, based on the description of the option in the first line.

#### Date/Time range for reporting:

Function: Selects for reporting only that data collected within a specified date and time range. The specified values are compared against the date and time in a given VCC run. Note that every VCC run has the same date and time values for each collected data.

Usage: Required.

Operand 1: Specify the minimum data collection time to be included in this report in the format ddmonyy:hh:mm:ss. Any data collected at or after this value will be included. Default value is 01JAN60:00:00:00 (January 01, 1960).

Operand 2: Specify the maximum data collection time to be included in this report in the format ddmonyy:hh:mm:ss. Any data collected at or before this value will be included. Default value is 31DEC59:23:59:59 (December 31, 2059).

Example: Include all data collected between 9 a.m. on March 4th 1991 and March 5th at midnight:

'04MAR91:09:00' - '05MAR91:23:59:59'

Enter the database ID for the database to be processed.

**Function:** Specifies the CA MICS Unit Database ID (DBID) from which the data for this report should be extracted. The value coded must specify a unit containing the VCA information area. You can enter a question mark (?) to obtain a list of the valid units for your installation. Check with your CA MICS Administrator if you have further questions or problems.

**Usage:** Required.

**Operand 1:** Specify the one-character Unit Database ID (DBID) that should be used to obtain data for this report. The default value can be specified as a MICF option.

**Operand 2:** Not used.

**Example:** Run a report using data from the San Francisco office. Data from that office resides in Unit Database ID (DBID) S:  
S

Enter the range of CA MICS File Cycles to be processed.

**Function:** Specifies the cycles of the CA MICS VCA information area to be used as the input data source for this report request. Typically, each cycle represents a snapshot of your DASD environment taken on a particular day. Cycle 01 would represent the most current snapshot, cycle 02 would represent a snapshot taken two days ago, etc.

**Usage:** Required.

**Operand 1:** Specify a numeric cycle number 01 through 99 to be used as the beginning cycle number to include in this report. The default value can be specified as a MICF option.

**Operand 2:** Specify a numeric cycle number 01 through 99 to be used as the ending cycle number to include in this report. If this is not specified, only the beginning cycle number

will be used. Specify this operand with caution, as it can cause duplicate data unless you supply a report date or specify a repetitive data option.

Example: Run a report using VCA data starting with cycle 01 and ending with cycle 05:  
01 - 05

Repetitive data processing option:

Function: The input data in the specified cycle(s) is selected according to the data's VCC run timestamp. Normally, VCC is run once a day against a specified set of the installation's DASD devices and the data collected is input to the CA MICS DAILY job. If your installation follows this procedure, then this option has limited use and can be skipped. This option has meaning only if each cycle of data in the VCA information area is comprised of data from more than one run of VCC. In this case, the data is grouped in each cycle according to the timestamp, that is kept with the data. This timestamp is the time at which the VCC collected the data.

Usage: Required.

Operand 1: Specify LATEST to select the most recent data in the specified cycle(s) (only the data collected during the most recent VCC run). Specify MERGE to select the most recent data associated with each unique data set or volume serial number. This allows for the case where the data contains many runs of the VCC data, each run containing data on a subset of the volumes in the configuration. The use of MERGE solves this problem by merging together several VCC runs and selecting the most recent occurrence for each unique data set and volume. Specify NONE to bypass testing for repetitive data. NONE is the default.

Operand 2: Not used.

Example: Include only data from the VCC run with the most current timestamp in this report:  
LATEST

Report Title (for first title of report):

Function: Specifies the title that will appear as the first title line on each report page printed by this inquiry.

Usage: Required.

Operand 1: Specify a character value that will appear as the first title line on each report page. Avoid using single quotes ('), double quotes (") or other special characters in the title value, as these can cause SAS errors or incorrect output. If your alternate title will not fit into the first operand area, it can be extended into the second area. If this is done, note that a blank space will always be inserted between the first and second operands. This problem can be minimized by coding leading blanks before the first operand value, so the break between the operands occurs where a blank is desired. The leading blanks will not appear in the title. The default title is CA MICS DASD MANAGEMENT REPORTING SYSTEM.

Operand 2: See Operand 1 above.

Example: Use an alternate report title for a 3390 DASD study that is being done: JUNE STUDY OF 3390 ACTIVITY

Select by DASD Device Type:

Function: Allows data sets and volumes to be included or excluded from reporting based on device type.

Usage: Optional.

Operand 1: Specify EQ if the DASD device type value must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those device types not equal to one of the values specified for operand 2.

Operand 2: Specify one or more device type values to be used for the comparison. If this operand is

left blank, no selection by device type will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value. Refer to data dictionary element DEVTYPE for a list of possible values.

Example: Include only 3380-K and 3390 DASD data in this report:  
EQ - 3380-K 3390\*

Select by Data Set Name:

Function: Allows data sets to be included or excluded from reporting based on the data set name. Refer to data dictionary element DSNAME for more information.

Usage: Optional.

Operand 1: Specify EQ if the data set name must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those data set names types not equal to one of the values specified for operand 2.

Operand 2: Specify one or more data set name values to be used for the comparison. If this operand is left blank, no selection by data set name will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value.

Example: Include only data sets with the prefix IMS and PROD. IMS:  
EQ - IMS\* PROD. IMS\*

Select by Data Set Organization:

Function: Allows data sets to be included or excluded from reporting based on their data set organization.

Usage: Optional.

Operand 1: Specify EQ if the data set organization must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those data set organizations not equal to one of the values

specified for operand 2.

Operand 2: Specify one or more data set organization values to be used for the comparison. If this operand is left blank, no selection by data set organization will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value. Refer to data dictionary element DSORG for possible values.

Example: Exclude data set organizations of DA and DAU from processing:  
NE - DA\*

Select by Date Last Used (ddmnyy):

Function: Allows data sets to be included or excluded from reporting based on the date they were last used. Each time a data set is opened for input or output, the last reference date for the data set is updated. This is collected by VCC and kept as VCA element DAAREFDT.

Usage: Optional.

Operand 1: Specify EQ if the value in the data must be equal to one of the operand 2 values. Specify NE if the data value must not equal one of the operand 2 values. Other values that can be specified include LT (data less than operand 2), GT (data greater than operand 2), LE (data less than or equal to operand 2), GE (data greater than or equal to operand 2), or R (data within the range specified by operand 2).

Operand 2: Specify one or more date values in the format ddmnyy to be used for the comparison. One or more values can appear if operand 1 is EQ or NE. Only one value should appear if operand 1 is LT, GT, LE, or GE. Two values should appear if operand 1 is R. If no values are specified, selection by date last used will not occur.

Example: Include all data sets used in May of 1990:  
R - 01MAY90 31MAY90

Select by Date of Creation (ddmnyy):

Function: Allows data sets to be included or excluded from reporting based on the date they were created. Refer to data dictionary element CREATEDT for more information.

Usage: Optional.

Operand 1: Specify EQ if the value in the data must be equal to one of the operand 2 values. Specify NE if the data value must not equal one of the operand 2 values. Other values that can be specified include LT (data less than operand 2), GT (data greater than operand 2), LE (data less than or equal to operand 2), GE (data greater than or equal to operand 2), or R (data within the range specified by operand 2).

Operand 2: Specify one or more date values in the format ddmonyy to be used for the comparison. One or more values can appear if operand 1 is EQ or NE. Only one value should appear if operand 1 is LT, GT, LE, or GE. Two values should appear if operand 1 is R. If no values are specified, selection by date of creation will not occur.

Example: Exclude all data sets created before 2008:  
GE - 01JAN08

Select by Free Tracks on Volume:

Function: Allows volumes to be included or excluded from reporting based on the number of free (unallocated) tracks present on the volume. Refer to data dictionary element VOASPCF for more information.

Usage: Optional.

Operand 1: Specify EQ if the value in the data must be equal to one of the operand 2 values. Specify NE if the data value must not equal one of the operand 2 values. Other values that can be specified include LT (data less than operand 2), GT (data greater than operand 2), LE (data less than or equal to operand 2), GE (data greater than or equal to operand 2), or R

(data within the range specified by operand 2).

Operand 2: Specify one or more numeric values to be used for the comparison. One or more values can appear if operand 1 is EQ or NE. Only one value should appear if operand 1 is LT, GT, LE, or GE. Two values should appear if operand 1 is R. If no values are specified, selection based on the number of free tracks will not occur.

Example: Include all volumes with more than 1000 tracks of unallocated space:  
GT - 1000

Select by Number of Extents:

Function: Allows data sets to be included or excluded from reporting based on the number of extents they have allocated. Refer to data dictionary element DAAEXTNO for more information.

Usage: Optional.

Operand 1: Specify EQ if the value in the data must be equal to one of the operand 2 values. Specify NE if the data value must not equal one of the operand 2 values. Other values that can be specified include LT (data less than operand 2), GT (data greater than operand 2), LE (data less than or equal to operand 2), GE (data greater than or equal to operand 2), or R (data within the range specified by operand 2).

Operand 2: Specify one or more numeric values to be used for the comparison. One or more values can appear if operand 1 is EQ or NE. Only one value should appear if operand 1 is LT, GT, LE, or GE. Two values should appear if operand 1 is R. If no values are specified, selection based on the number of extents will not occur.

Example: Include only those data sets that have no extents allocated:  
EQ - 0

Select by Number of Tracks Allocated:

Function: Allows data sets to be included or excluded from reporting based on the number of tracks they have allocated. Refer to data dictionary element DAASPACA for more information.

Usage: Optional.

Operand 1: Specify EQ if the value in the data must be equal to one of the operand 2 values. Specify NE if the data value must not equal one of the operand 2 values. Other values that can be specified include LT (data less than operand 2), GT (data greater than operand 2), LE (data less than or equal to operand 2), GE (data greater than or equal to operand 2), or R (data within the range specified by operand 2).

Operand 2: Specify one or more numeric values to be used for the comparison. One or more values can appear if operand 1 is EQ or NE. Only one value should appear if operand 1 is LT, GT, LE, or GE. Two values should appear if operand 1 is R. If no values are specified, selection based on the number of allocated tracks will not occur.

Example: Exclude those data sets occupying less than 500 tracks of DASD space:  
GE - 500

Select by Number of Tracks Used:

Function: Allows data sets to be included or excluded from reporting based on the number of tracks actually being used for data storage. Refer to data dictionary element DAASPACU for more information.

Usage: Optional.

Operand 1: Specify EQ if the value in the data must be equal to one of the operand 2 values. Specify NE if the data value must not equal one of the operand 2 values. Other values that can be specified include LT (data less than operand 2), GT (data greater than operand 2), LE (data

less than or equal to operand 2), GE (data greater than or equal to operand 2), or R (data within the range specified by operand 2).

Operand 2: Specify one or more numeric values to be used for the comparison. One or more values can appear if operand 1 is EQ or NE. Only one value should appear if operand 1 is LT, GT, LE, or GE. Two values should appear if operand 1 is R. If no values are specified, selection based on the number of used tracks will not occur.

Example: Include only those data sets using more than 500 tracks, but less than 1000:  
R - 501 999

Select by Percent Volume Free:

Function: Allows volumes to be included or excluded from reporting based on the percentage of space on the volume that remains available for new data set allocations. Refer to data dictionary element VOAPCSPF for more information.

Usage: Optional.

Operand 1: Specify EQ if the value in the data must be equal to one of the operand 2 values. Specify NE if the data value must not equal one of the operand 2 values. Other values that can be specified include LT (data less than operand 2), GT (data greater than operand 2), LE (data less than or equal to operand 2), GE (data greater than or equal to operand 2), or R (data within the range specified by operand 2).

Operand 2: Specify one or more numeric values to be used for the comparison. One or more values can appear if operand 1 is EQ or NE. Only one value should appear if operand 1 is LT, GT, LE, or GE. Two values should appear if operand 1 is R. If no values are specified, selection based on the percentage of free space will not occur.

Example: Include only those volumes with less than 10

percent free space:  
LT - 10

Select by Percentage of Space Used:

Function: Allows data sets to be included or excluded from reporting based on the percentage of space in the data set actually being used to store data. If a data set has allocated 100 tracks of space but is only using 25 tracks, the percentage of space used is 25. Refer to data dictionary element DAAPCPU for more information.

Usage: Optional.

Operand 1: Specify EQ if the value in the data must be equal to one of the operand 2 values. Specify NE if the data value must not equal one of the operand 2 values. Other values that can be specified include LT (data less than operand 2), GT (data greater than operand 2), LE (data less than or equal to operand 2), GE (data greater than or equal to operand 2), or R (data within the range specified by operand 2).

Operand 2: Specify one or more numeric values to be used for the comparison. One or more values can appear if operand 1 is EQ or NE. Only one value should appear if operand 1 is LT, GT, LE, or GE. Two values should appear if operand 1 is R. If no values are specified, selection based on the percentage used space will not occur.

Example: Include only those data sets using 50 percent or less of their allocated space:  
LE - 50

Select by SMS Data Class:

Function: Allows data sets to be included or excluded from reporting based on their assigned DFSMS data class. Refer to data dictionary element STORDATC for more information.

Usage: Optional.

Operand 1: Specify EQ if the data class name must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those data sets with data classes not equal to one of the values specified for operand 2.

Operand 2: Specify one or more data class names to be used for the comparison. If this operand is left blank, no selection by data class name will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value. A keyword of NULL can also be specified to select data sets with no (blank) data class assigned.

Example: Include data sets with data class names SPECIAL, with the prefix TEST, or with no data class:  
EQ - SPECIAL TEST\* NULL

Select by SMS Management Class:

Function: Allows data sets to be included or excluded from reporting based on their assigned DFSMS management class. Refer to data dictionary element STORMGTC for more information.

Usage: Optional.

Operand 1: Specify EQ if the management class name must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those data sets with management classes not equal to one of the values specified for operand 2.

Operand 2: Specify one or more management class names to be used for the comparison. If this operand is left blank, no selection by management class name will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value. A keyword of NULL can also be specified to select data sets with no (blank) management class assigned.

Example: Exclude data sets with management class names with the prefix NOMIG:  
NE - NOMIG\*

Select by SMS Storage Class:

Function: Allows data sets to be included or excluded from reporting based on their assigned DFSMS storage class. Refer to data dictionary element STORSTGC for more information.

Usage: Optional.

Operand 1: Specify EQ if the storage class name must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those data sets with storage classes not equal to one of the values specified for operand 2.

Operand 2: Specify one or more storage class names to be used for the comparison. If this operand is left blank, no selection by storage class name will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value. A keyword of NULL can also be specified to select data sets with no (blank) storage class assigned.

Example: Include data sets with storage class names STANDARD and FAST:  
EQ - STANDARD FAST

Select by SMS Storage Group:

Function: Allows data sets and volumes to be included or excluded from reporting based on their assigned DFSMS storage group. Refer to data dictionary element STORNAME for more information.

Usage: Optional.

Operand 1: Specify EQ if the storage group name must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those data sets with storage groups not equal to one of the values specified for operand 2.

Operand 2: Specify one or more storage group names to be used for the comparison. If this operand is

left blank, no selection by storage group name will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value. A keyword of NULL can also be specified to select data sets with no (blank) storage group class assigned.

Example: Exclude data sets with any storage group name prefixed by the characters TS0:  
NE - TS0\*

Select by specific System (SYSID):

Function: Allows data sets and volumes to be included or excluded from reporting based on their System Identifier. Refer to data dictionary element SYSID for more information.

Usage: Optional.

Operand 1: Specify EQ if the system identifier must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those data sets with system identifiers not equal to one of the values specified for operand 2.

Operand 2: Specify one or more system identifiers to be used for the comparison. If this operand is left blank, no selection by system identifier will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value.

Example: Include data sets with system identifier PROD, plus any with the prefix IMS:  
EQ - PROD IMS\*

Select by VCA Account Field:

Function: Allows data sets to be included or excluded from reporting based on the contents of any of the VCA Account Fields. Up to 9 different account fields can be defined within your VCA data. The contents of these fields are specific to your installation, but can help summarize your data by location, department, division, user, etc. Contact your CA MICS Administrator if you are unsure of the

structure and format of your accounting fields. Note that all 9 fields may not be defined at your installation.

Usage: Optional.

Operand 1: Specify EQ if the account field contents must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those data sets with account field data not equal to one of the values specified for operand 2.

Operand 2: Specify one or more values to be used for the comparison. If this operand is left blank, no selection by account field contents will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value.

Example: Include data sets with values of PAY, TEST, or SYSTEM in this account field, or with a prefix of CICS:

EQ - PAY CICS\* TEST SYSTEM

Select by Volume Fragmentation Index:

Function: Allows volumes to be included or excluded from reporting based on the value of their fragmentation index. This index is used as a measure of the fragmentation of the free space on the volume. Refer to data dictionary element VOAFRAGX for more information.

Usage: Optional.

Operand 1: Specify EQ if the value in the data must be equal to one of the operand 2 values. Specify NE if the data value must not equal one of the operand 2 values. Other values that can be specified include LT (data less than operand 2), GT (data greater than operand 2), LE (data less than or equal to operand 2), GE (data greater than or equal to operand 2), or R (data within the range specified by operand 2).

Operand 2: Specify one or more numeric values to be used for the comparison. One or more values can appear if operand 1 is EQ or NE. Only one

value should appear if operand 1 is LT, GT, LE, or GE. Two values should appear if operand 1 is R. If no values are specified, selection based on the fragmentation index will not occur.

Example: Include only those volumes with fragmentation index values between .500 and .900:  
R - .500 .900

Select by Volume Serial Number:

Function: Allows data sets and volumes to be included or excluded from reporting based on the serial number of the volume. Refer to data dictionary element VOLSER for more information.

Usage: Required for the Volume Track Map Report. Optional for all other reports.

Operand 1: Specify EQ if the volume serial number must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those volumes with serial numbers not equal to one of the values specified for operand 2.

Operand 2: Specify one or more values to be used for the comparison. If this operand is left blank, no selection by volume serial number will occur. Specify an asterisk (\*) as the last character to indicate a prefix or short value.

Example: Include all volumes in this report with the serial number prefixes TSO, PAY, and SPOOL:  
EQ - PAY\* TSO\* SPOOL\*

Select SMS Managed Data Sets:

Function: Allows data sets to be included or excluded from reporting based on their DFSMS management status. Data sets residing on DFSMS controlled volumes are known as managed data sets, as all future allocation and movement activities for those data sets are managed by DFSMS. Refer to data dictionary element DAASMSID for more information.

Usage: Optional.

Operand 1: Specify YES if only DFSMS managed data sets should be included in the reporting. Specify NO if only unmanaged data sets should be included. Leave this operand blank to cause no selection based on DFSMS management status to occur.

Operand 2: Not used.

Example: Include only DFSMS managed data sets in this report:  
YES

Select SMS Volume Ownership Status:

Function: Allows volumes to be included or excluded from reporting based on their DFSMS management status. Volumes fully managed by DFSMS are known as Converted volumes, while volumes in the process of being converted are said to be in Initial status. Refer to data dictionary element VOASMSFG for more information.

Usage: Optional.

Operand 1: Specify EQ if the volume DFSMS status must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those volumes with DFSMS status not equal to one of the values specified for operand 2.

Operand 2: Specify INITIAL to select or exclude those volumes that are in DFSMS Initial status. Specify CONVERTED to select or exclude those volumes that are in DFSMS converted status. Leave this operand blank to cause no selection based on DFSMS volume status to occur.

Example: Include only those volumes in this report that are not managed by DFSMS:  
NE - INITIAL CONVERTED

Select Volume Mount Attribute:

Function: Allows volumes to be included or excluded from reporting based on their MVS mount attribute.

Refer to data dictionary element VOAMOUNT for more information.

Usage: Optional.

Operand 1: Specify EQ if the MVS mount attribute must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those volumes with mount attributes not equal to one of the values specified for operand 2.

Operand 2: Specify RESIDENT to select or exclude those volumes mounted as Resident. Specify RESERVED to select or exclude those volumes mounted as Reserved. Specify REMOVABLE to select or exclude those volumes mounted as Removable. Leave this operand blank to cause no selection based on MVS mount attribute to occur.

Example: Include only those volumes in this report that are mounted with a status of Removable.  
EQ - REMOVABLE

Select Volume Usage Attribute:

Function: Allows volumes to be included or excluded from reporting based on their MVS usage attribute. Refer to data dictionary element VOAUSE for more information.

Usage: Optional.

Operand 1: Specify EQ if the MVS usage attribute must be equal to one of the operand 2 values in order to be included in this report. Specify NE to include only those volumes with usage attributes not equal to one of the values specified for operand 2.

Operand 2: Specify PRIVATE to select or exclude those volumes with Private usage status. Specify PUBLIC to select or exclude those volumes with Public usage status. Specify STORAGE to select or exclude those volumes with Storage usage status. Leave this operand blank to cause no selection based on MVS usage attribute to occur.

Example: Exclude those volumes from this report that  
are mounted with a usage attribute of Public:  
NE - PUBLIC

### 3.8.3 Macro Variable Override Options

The Data Set Size Analysis, the Data Set Fragmentation Analysis, and the Data Set Last Reference Analysis sections of the Overview report display distributions of data sets within a set of predefined categories. The values associated with these categories are adjustable by the user. The Exception Analysis section reports on the five most critical exceptions. The exception numbers can also be specified by the user. A set of macro variables has been defined which specify the lower bound values for each of the categories and the number of the most significant exceptions. The macro variables are defined in the VCARPTU1 member of sharedprefix.MICS.SOURCE. The names and descriptions of the macro variables are shown in Figure 3-8.

Report	SOURCE Member Name	Macro Name	Default Values	Macro Description
Overview	VCARPTU1	_EXC1 thru _EXC5	'09001' '09002' '09003' '09004' '09005'	The five most significant exception codes contained in the Exception Report
		_EXCPRE1 thru _EXCPRE5		The prefix strings that are used to make the exception descriptions more readable on the Exception Report
		_DSALL1 thru _DSALL5	2000 1000 500 250 100	The five lower bound TRACK values for the allocated tracks category used in the Data Set Size Analysis
		_REFDT1 thru _REFDT5	13 10 7 4 1	The five lower bound MONTH values for the last reference category used in the Data Set Last Reference Analysis
		_DSEXT1 thru _DSEXT4	13 9 5 2	The four lower bound EXTENT values for the allocated extent category used in the Data Set Fragmentation Analysis

Figure 3-8. Report User Macro Definitions

### 3.8.4 Sample Inquiry Execution

This section will illustrate the procedure used to select and execute one of the MICF standard analysis inquiries supplied with the CA MICS Space Analyzer product.

In this example, the user wishes to execute the Volume Summary Report to list volumes that may have potential space problems due to volume fragmentation.

The user first establishes a TSO session, invokes ISPF, and selects the CA MICS Workstation Facility (MWF) option. The exact procedure for doing this will vary for each site, so contact your CA MICS Administrator if you are unsure of this procedure.

From the MWF primary menu, select the MICF option. From the primary MICF menu, select the Database Inquiries option. At this point, you should see a screen similar to that shown in Figure 3-9.

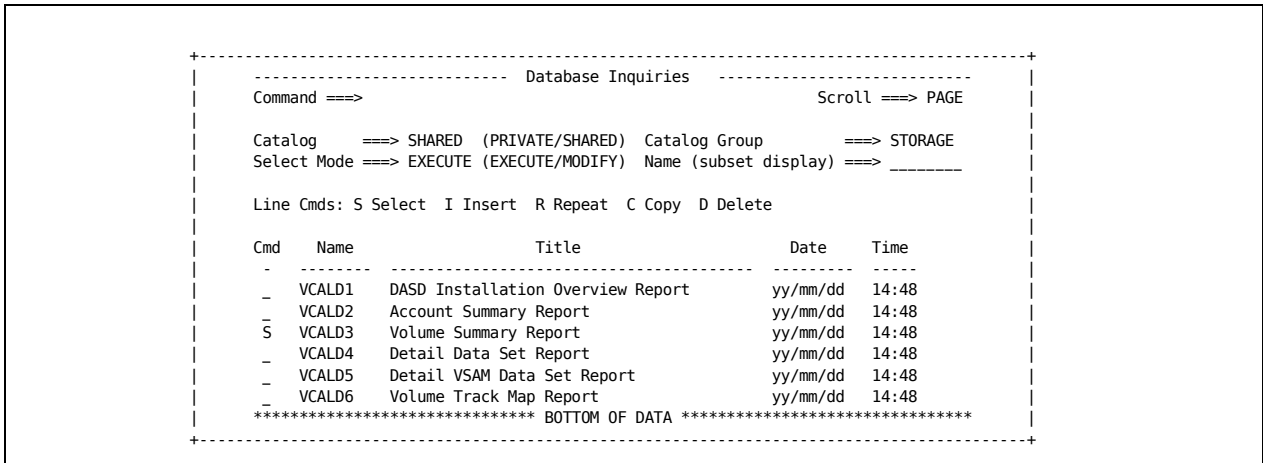


Figure 3-9. MICF Inquiry Selection Screen

If the screen you see is not similar to the sample above, make sure the Catalog, Catalog Group, and Select Mode are identical to that shown. Note also that other CA MICS storage products may have added inquiries to the STORAGE Catalog Group, so you may see inquiries other than those shown in Figure 3-9. If this occurs, use the ISPF scroll keys to locate the inquiries prefixed with VCA.

In the example above, a line command of 'S' (Select), has been placed to the left of the report that is being selected for execution.

When the Volume Summary Report is selected for execution, you may be shown an additional screen that allows you to override execution options, such as whether you desire batch or interactive execution, and the format of the Job card(s) for batch submission. This screen is not shown in this example. If it appears, simply follow the instructions on the screen and complete the options.

After the Volume Summary Report has been selected for execution, the Execution-Time Parameters screen will appear. An example of this screen is shown in Figure 3-10.

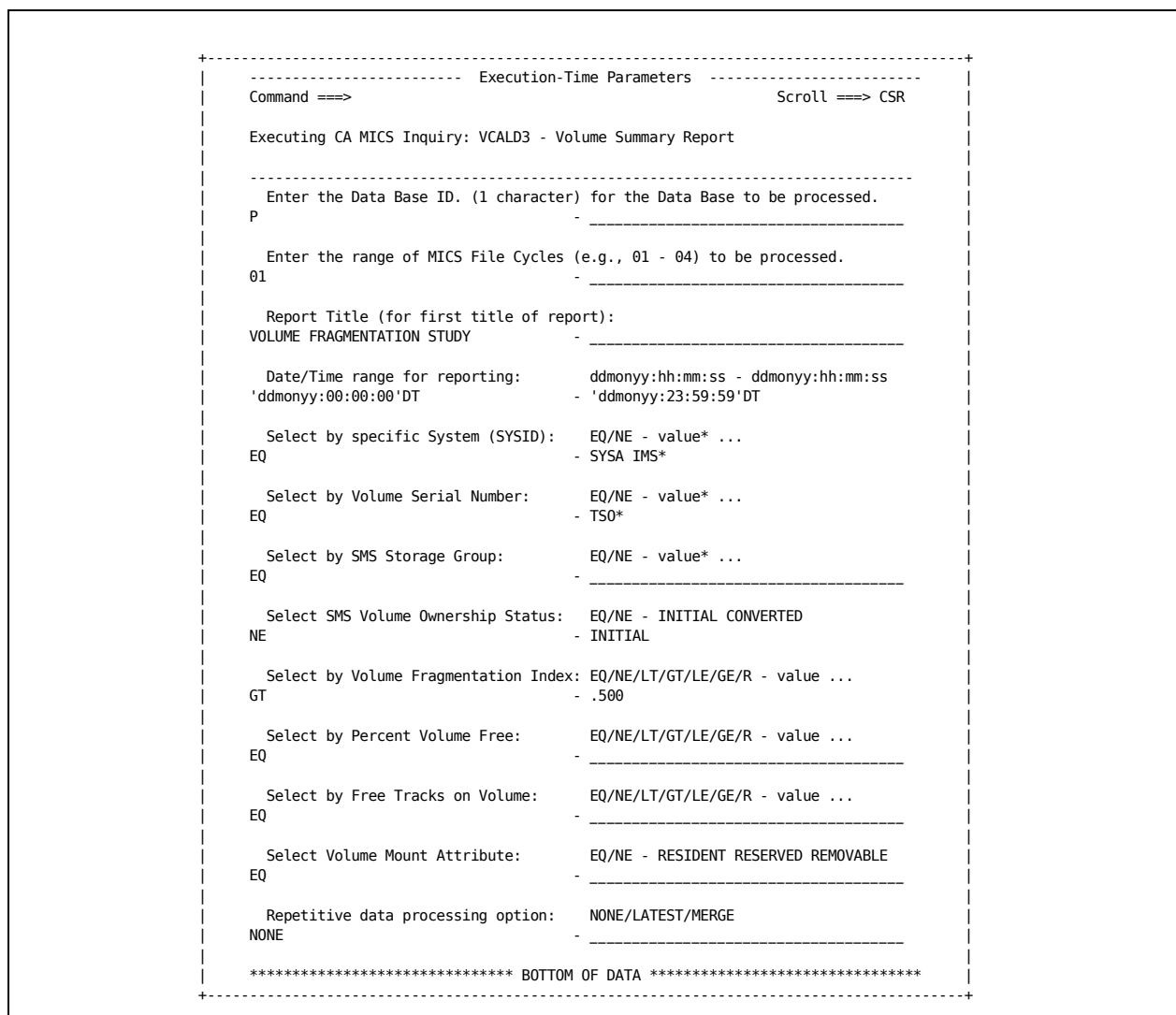


Figure 3-10. Sample Execution Parameter Screen

The Execution-Time Parameters screen is then used to enter the parameters that will control the execution of the inquiry and the resulting output that will be produced. The ISPF scroll keys can be used to scroll up and down through the various options.

Based on the example shown in Figure 3-10, the following parameters have been entered by this user:

- Input data should be obtained from Cycle 01 of Data Base Unit P.
- The default title was changed to VOLUME FRAGMENTATION STUDY.
- The default report Date/Time range was not changed.
- Data will only be selected if its System Identifier (SYSID) is SYSA, or has a prefix of IMS.
- Only those volumes will be accepted that have a volume serial number with the prefix TS0.
- Volumes will be excluded if they are in DFSMS Initial status (in the process of being converted to DFSMS management).
- Only volumes with a fragmentation index greater than .500 will be accepted for reporting. This will only select volumes that are starting to have space fragmentation problems.

Once the specified parameters are acceptable, the user should use the ISPF END key to continue report execution. This should generate a report meeting the specifications requested by the user.

### 3.8.5 MICF Production Reporting Support

The standard analysis reports supplied with the CA MICS Space Analyzer are fully supported by the MICF Production Reporting facility. The inquiries may be scheduled to execute automatically as part of normal CA MICS processing.

This facility makes it easy to schedule the regular production of reports that can be used to monitor your storage resources. Anomalies detected in these regular reports can then be further investigated using other standard analysis reports run under MICF.

The Volume Track Map Report may have to be modified when used with the MICF Production Reporting facility. This is because, by default, the report will only select volumes with a serial number prefix of "TS0" for reporting. In most cases this is not a problem, as this report is normally used for specific problem research and not as part of regular production reporting. If you desire to use this report in a production reporting environment, you may have to copy the inquiry, edit the default value to meet your standards, and then specify the modified inquiry as input to the MICF Production Reporting facility.

# Chapter 4: EXCEPTIONS

---

The Space Analyzer provides the necessary detail data to specifically identify DASD exception conditions. These exceptions are used to assist in the management of the DASD configuration.

The following exception conditions are reported:

Number	Severity	Management Area	Exception Description
09001	Impacting	Availability	Available VTOC Space Shortage
09002	Impacting	Availability	Available Volume Space Shortage
09003	Warning	Performance	High Alternate Track Usage
09004	Warning	Performance	Volume Free Space is Fragmented
09005	Warning	Performance	Data Set has too Many Extents
09006	Warning	Performance	VSAM Data Set has too Many Extents
09007	Warning	Performance	Fixed-length PS or PO Data Set is Unblocked
09008	Warning	Performance	Multi-extent DSORG=DA Data Set
09010	Warning	Performance	Data Set Using Less Than 'n' % of Allocated Space
09012	Warning	Performance	VTOC Reorganization Recommended
09013	Impacting	Availability	VTOC in Error, Requires Restructure (DIRF set)
09014	Impacting	Availability	Tracks not Accounted for in VTOC
09015	Impacting	Performance	VTOC Does Not End on Cylinder Boundary
09016	Warning	Performance	Large Data Set not Cylinder Allocated
09017	Impacting	Performance	VTOC is not in Indexed Format
09018	Critical	Reliability	Indexed VTOC Potentially in Error
09019	Warning	Availability	Invalid DS1LSTAR Value

Figure 4-1. CA MICS Space Analyzer Exceptions

This section contains the following topics:

[4.1 Exception Process Overview](#) (see page 78)

[4.2 Setting Exception Values](#) (see page 81)

[4.3 Detailed Exception Descriptions](#) (see page 82)

## 4.1 Exception Process Overview

The CA MICS exception process consists of the individual exceptions, a set of standard reports, CA MICS Information Center Facility (MICF) inquiries, an exception test routine, and an exception value analysis routine. Each of these is described below.

### EXCEPTIONS

An exception is the occurrence of an event which merits visibility and attention. It may be an occurrence that is a distinct problem (for example, CICS abended at 2:00 p.m.) or one that may be a problem and requires further research (for example, a TSO user overloaded the system from 1:00 to 1:30 p.m.), or it may represent a standard, security, or audit violation (for example, user XYZ is not authorized to use PDZAP and was detected using it seven times yesterday).

Because the volume of exception occurrences can be quite large, CA MICS provides means to categorize, aggregate, consolidate, and prioritize them to meet your needs. Each exception has:

- o An exception number for unique definition
- o A severity level to signify degree of importance
- o A management area to identify area of responsibility

### STANDARD EXCEPTION REPORTS

The standard exception reports provide a concise, integrated method for problem reporting. The following reports can be produced as part of the CA MICS DAILY job:

- o Exception Management Overview Report
- o Severity Level Exception Summary Report
- o Management Area Exception Summary Report

You control which reports are produced via the REPORT EXCEPTIONS statement in prefix.MICS.PARMS(EXECDEF). For more information, refer to Section 2.3.5 of the CA MICS Planning, Installation, Operation, and Maintenance Guide (PIOM).

Two additional standard reports can be produced as required to provide the necessary background detail to effectively analyze reported exceptions. The two reports are:

- o Full Exception Detail Report
- o Short Exception Detail Report

#### MICF INQUIRIES

The catalog group EXCEPT that is shipped with CA MICS contains a number of standard MICF inquiries that you can use to report exception conditions. In the following inquiry list, graphic reports whose value of x is C produce color graphic reports using SAS/GRAPH. If the value of x is P, printer graphic reports are produced without using SAS/GRAPH.

- o BASxM1 - Monthly Exception Summary Report
- o BASxM2 - Monthly Mgmt. Area Exception Summary Report
- o BASxM3 - Monthly Info. Area Exception Summary Report
- o BASxM4 - Monthly Exception Management Overview Report
- o BASxM5 - Monthly Info. Area Exception Overview Report
- o BASxW1 - Weekly Exception Summary Report
- o BASxW2 - Weekly Mgmt. Area Exception Summary Report
- o BASxW3 - Weekly Info. Area Exception Summary Report

The following inquiries that produce printed reports are also available:

- o BASLD2 - Daily Severity Level Exception Summary Report
- o BASLD3 - Daily Mgmt. Area Exception Summary Report
- o BASLD4 - Daily Short Exception Detail Report
- o BASLD5 - Daily Full Exception Detail Report
- o BASLD6 - Daily Exception Ranking Report
- o BASLM6 - Monthly Exception Ranking Report

These standard inquiries have execution-time parameter selection that permits you to report on a subset of the exceptions. For example, inquiry BASLD5 allows selection on SYSID, Information Area, Management Area, Severity Level, and other criteria.

#### EXCEPTION TEST ROUTINE

Each CA MICS product has an exception test routine that is invoked in the DAY200 step of the CA MICS DAILY job. You control which routines are invoked using the CREATE EXCEPTIONFILES statement in prefix.MICS.PARMS(EXECDEF). For more information on EXECDEF, refer to Section 2.3.5 of the CA MICS Planning, Installation, Operation, and Maintenance Guide (PIOM).

An exception test routine written in the SAS language, defines the exception and tests to determine whether or not the exception condition is present in the data being processed. The distributed exception test routine for each CA MICS product is contained in `sharedprefix.MICS.SOURCE(DYcccEXC)`, where `ccc` is the product identifier. When the `DAY200` step invokes the test routines, it does so by `%INCLUDEing` the `DYcccEXC` member from `prefix.MICS.USER.SOURCE`. As distributed, that member then `%INCLUDEs` the member from `sharedprefix.MICS.SOURCE`. For more information on `DYcccEXC`, refer to Section 4.2 of this guide, *Setting Exception Values*.

### EXCEPTION VALUE ANALYSIS ROUTINE

Each CA MICS product has an exception value analysis (EVA) routine that you can use to help determine values for the exception conditions. The EVA routine extracts information from the CA MICS database. Descriptive statistics for the values of variables used in exception tests are printed in the Exception Value Analysis Report. The EVA routine for each product is stored in `sharedprefix.MICS.SOURCE(cccEVA)`. The job control to execute the EVA process is contained in `prefix.MICS.CNTL(cccEVA)`.

## 4.2 Setting Exception Values

Exception test routines contain tests that determine the conditions and definitions that classify the exception for reporting and analysis. The following is a sample exception test:

```

/*-----*/
/* 09001 */
/* AVAILABLE VTOC SPACE SHORTAGE */
/*-----*/

IF VOADSCB0 LT 25

    THEN DO;
        EXCCODE = '09001'; SEVERITY= 'I'; MGMTAREA= 'AVAILABILITY';
        EXCDESC1 = 'AVAILABLE VTOC DSCB SHORTAGE';
        EXCDESC2 = 'AVAILABLE DSCB COUNT=' || PUT(VOADSCB0,3.)
                || ', VOL=' || VOLSER;
        LINK HIT;
    END;

```

This exception test is processed for each observation that is in the latest cycle of the DETAIL Volume Allocation File (VCAVOA). The test is positive only when the number of Format 0 DSCBs (VOADSCB0) is less than 25.

When the test is positive, the exception is categorized by providing the appropriate values for EXCCODE, SEVERITY, and MGMTAREA. EXCDESC1 provides a consistent label for the exception. EXCDESC2 provides variable information for the conditions that caused the test to be positive. The LINK HIT statement invokes a routine that causes the exception condition eventually to be written to the Exception Activity File (ADMEXC) for later processing by the standard reports or MICF inquiries.

For most organizations, the exception limit used here would either cause too many exceptions or too few exceptions to be reported, which defeats one of the purposes of exception processing: to report on "out of the ordinary" conditions that warrant attention. Therefore, each organization must determine and set its own exception value.

Furthermore, the values for some exception conditions should be determined uniquely for different environments within a single organization. For example, a system paging rate that would be excessive during the nightly batch processing may be normal during the daytime hours, which have heavy interactive

usage. Also, different processors can support different paging rates. Numbers used as exception values should not be defined without some analysis of installation history, performance, and user requirements.

The CA MICS database provides a most useful information base for analyzing historical performance. The exception value analysis routine uses the CA MICS database to produce a statistical analysis of the values used in the standard exception tests distributed with the products.

Using the results of this analysis along with your installation's internal political, security, or standards policies, industry publications, and your own analysis of data element behavior, you can determine how to modify each exception test for your environment. You can also modify the exception test routine to add your own tests for additional exception conditions not detected by the standard tests delivered with the product.

## 4.3 Detailed Exception Descriptions

This section describes each of the standard exceptions for the CA MICS Space Analyzer. The exceptions are organized numerically and appear sequentially starting with exception 09001.

The format provides the title, number, and a statement on the purpose, rationale, and definition for each exception to give you some insight into the meaning and use of the information. Finally, the description contains the SAS code for the exception, with an explanation on modifying the exception threshold values.

## 09001: Available VTOC DSCB Shortage

FILE: VOA Volume Allocation File  
 SAS FILE NAME: DETAIL.VCAVOA01  
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Impacting (SEVERITY='I')  
 MANAGEMENT AREA: Availability (MGMTAREA='AVAILABILITY')

PURPOSE: Identifies when the volume's VTOC is short on available Format 0 DSCBs.

RATIONALE: It is important that there be a sufficient number of available DSCBs in the VTOC so that new data sets can be allocated and old data sets can be extended.

DEFINITION: This exception is detected when the count of available DSCBs (VOADSCB0) drops below the specified minimum.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/
/*                                          */
/* 09001                                          */
/* AVAILABLE VTOC SPACE SHORTAGE                */
/*                                          */
/*-----*/
IF VOADSCB0 LT #dscb0
THEN DO;
  EXCCODE='09001'; SEVERITY='I'; MGMTAREA='AVAILABILITY';
  EXCDDESC1='AVAILABLE VTOC DSCB SHORTAGE';
  EXCDDESC2='AVAILABLE DSCB COUNT=' || PUT(VOADSCB0,3.) ||
    ', VOL=' || VOLSER;
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: The user should modify the #dscb0 value according to the following conventions:

#dscb0 - The minimum acceptable count of available Format 0 DSCBs. A minimum count of 25 available Format 0 DSCBs would appear as :

```
VOADSCB0 LT 25
```

## 09002: Available Volume Space Shortage

FILE: VOA Volume Allocation File  
SAS FILE NAME: DETAIL.VCAVOA01  
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Impacting (SEVERITY='I')  
MANAGEMENT AREA: Availability (MGMTAREA='AVAILABILITY')

PURPOSE: Identifies when the volume's space available for allocation falls below a specified level.

RATIONALE: It is important that there be a sufficient amount of available space on a volume so that new data sets can be allocated and old data sets can be extended.

DEFINITION: This exception is detected when the amount of available space (VOASPACF) drops below the specified minimum number of tracks.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/  
/* */  
/* 09002 */  
/* AVAILABLE VOLUME SPACE SHORTAGE */  
/* */  
/*-----*/  
IF VOASPACF LT #freesp  
THEN DO;  
  EXCCODE='09002'; SEVERITY='I'; MGMTAREA='AVAILABILITY';  
  EXCDDESC1='AVAILABLE VOLUME SPACE SHORTAGE';  
  EXCDDESC2='NUMBER OF FREE TRACKS=' || PUT(VOASPACF,3.) ||  
    ', VOL=' || VOLSER;  
  LINK HIT;  
END;
```

THRESHOLD MODIFICATION: The user should modify the #freesp value according to the following conventions:

#freesp - The minimum acceptable amount of space available for allocation, specified in tracks. A minimum amount of 500 free tracks would appear as:

```
VOASPACF LT 500
```

## 09003: High Alternate Track Usage

FILE: VOA Volume Allocation File  
 SAS FILE NAME: DETAIL.VCAVOA01  
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Warning (SEVERITY='W')  
 MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies when the number of assigned alternate tracks on a volume exceeds a specified level.

RATIONALE: DASD performance can be significantly degraded when data is stored on an alternate track due to a defective primary track. The extra disk access along with the necessity to reposition the access mechanism will result in increased data access times. Also, high assignments of alternate tracks can be an early indication of a failing disk assembly.

DEFINITION: This exception is detected when the count of assigned alternate tracks (VOAALNTK) exceeds the specified maximum.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/
/*                                          */
/* 09003                                          */
/* HIGH ALTERNATE TRACK USAGE                    */
/*                                          */
/*-----*/
IF VOANOATK GT #usedalt
THEN DO;
  EXCCODE='09003'; SEVERITY='W'; MGMTAREA='PERFORMANCE';
  EXCDDESC1='HIGH ASSIGNMENT OF ALTERNATE TRACKS';
  EXCDDESC2='ALTERNATE TRACKS USED=' || PUT(VOANOATK,3.) ||
    ', VOL=' || VOLSER;
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: The user should modify the #usedalt value according to the following conventions:

#usedalt - The maximum acceptable number of assigned alternate tracks. A maximum count of 5 would appear

as:

VOANOATK GT 5

## 09004: Volume Free Space is Fragmented

FILE: VOA Volume Allocation File  
 SAS FILE NAME: DETAIL.VCAVOA01  
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Warning (SEVERITY='W')  
 MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies when the number of areas of free space on a volume exceeds a specified level.

RATIONALE: Allocation of data sets on a volume can be impacted by the level of free space fragmentation. The probability of allocating large primary requests in a single extent can be greatly reduced, resulting in the allocation of data sets over several non-contiguous extents. This can result in poorer performance for the data sets affected.

DEFINITION: This exception is detected when the computed fragmentation index (VOAFRAGX) exceeds the specified maximum.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/
/*                                     */
/* 09004                               */
/* VOLUME FREE SPACE IS FRAGMENTED    */
/*                                     */
/*-----*/
IF VOAFRAGX GE #fragx
THEN DO;
  EXCCODE='09004'; SEVERITY='W'; MGMTAREA='PERFORMANCE';
  EXCDESC1='VOLUME FREE SPACE IS FRAGMENTED';
  EXCDESC2='FRAGMENTATION INDEX=' || PUT(VOAFRAGX,5.3) ||
    ', VOL=' || VOLSER;
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: The user should modify the #fragx value according to the following conventions:

#fragx - The maximum acceptable index of fragmentation on a volume. Acceptable values are between 0 and 1. A maximum count of .700 would appear as:

```
VOAFRAGX GE .700
```

## 09005: Data Set has Too Many Extents

FILE: DAA Data Set Allocation File  
SAS FILE NAME: DETAIL.VCADAA01  
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)  
  
SEVERITY: Warning (SEVERITY='W')  
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies when the number of data set extents exceeds a specified level.

RATIONALE: Access to a data set that expands into many extents can be less efficient than if the data set were allocated in a single extent because of the increased overhead required to reposition the data volume to each different extent. And because a data set is limited to 16 extents, processing will be aborted if an attempt is made to expand a data set that already has allocated 16 extents.

DEFINITION: This exception is detected when the number of data set extents (DAAEXTNO) exceeds the specified maximum.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/  
/* */  
/* 09005 */  
/* DATA SET HAS TOO MANY EXTENTS */  
/* */  
/*-----*/  
IF DAAEXTNO GE extent#  
THEN DO;  
  EXCCODE='09005'; SEVERITY='W'; MGMTAREA='PERFORMANCE';  
  EXCDESC1='DATA SET HAS TOO MANY EXTENTS. EXTENT COUNT=' ||  
    PUT(DAAEXTNO,3.);  
  EXCDESC2='VOL=' || VOLSER || ', DSN=' || DSNAME;  
  LINK HIT;  
END;
```

THRESHOLD MODIFICATION: The user should modify the extent# value according to the following conventions:

extent# - The maximum acceptable number of extents for a data set. A maximum value of 14 would appear as:

```
DAAEXTNO GE 14
```

## 09006: VSAM Data Set has Too Many Extents

FILE: DAA Data Set Allocation File  
 SAS FILE NAME: DETAIL.VCADAA01  
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Warning (SEVERITY='W')  
 MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies when the number of data set extents exceeds a specified level.

RATIONALE: Access to a data set that expands into many extents can be less efficient than if the data set were allocated in a single extent due to the increased overhead required to reposition the data volume to each different extent. Also, because a VSAM data set is limited to 128 extents, processing will be aborted if an attempt is made to expand a data set that already has allocated 128 extents.

DEFINITION: This exception is detected when the number of data set extents (DAAEXTNO) exceeds the specified maximum.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/
/*                                     */
/* 09006                               */
/* VSAM DATA SET HAS TOO MANY EXTENTS */
/*                                     */
/*-----*/
IF DAAEXTNO GE extent#
THEN DO;
  EXCCODE='09006'; SEVERITY='W'; MGMTAREA='PERFORMANCE';
  EXCDESC1='DATA SET HAS TOO MANY EXTENTS. EXTENT COUNT=' ||
    PUT(DAAEXTNO,3.);
  EXCDESC2='VOL=' || VOLSER || ', DSN=' || DSNAME;
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: The user should modify the extent# value according to the following conventions:

extent# - The maximum acceptable number of extents for a data set. A maximum value of 115 would appear as:

```
DAAEXTNO GE 115
```

## 09007: Fixed Length PO or PS Data Set is Unblocked

FILE: DAA Data Set Allocation File  
SAS FILE NAME: DETAIL.VCADAA01  
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Warning (SEVERITY='W')  
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies unblocked data set that are candidates for a more efficient use of space.

RATIONALE: A blocked data set can more efficiently use the DASD space allocated to it than an unblocked data set because of the overhead associated with each block of data stored on a DASD device. Increasing the block size reduces the amount of overhead associated with each record in the data, resulting in more records being stored per track of DASD space. Partitioned and sequential data sets that have fixed length records are likely candidates for blocking.

DEFINITION: This exception is detected when a data set's organization (DSORG) is either partitioned (PO) or sequential (PS), the record format (RECFM) is fixed (F), and the logical record length (LRECL) equals the block size (BLKSIZE).

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/  
/* */  
/* 09007 */  
/* FIXED LENGTH PO OR PS DATA SET IS UNBLOCKED */  
/* */  
/*-----*/  
IF (DSORG = 'PO' OR DSORG = 'PS') AND RECFM = 'F' AND  
    BLKSIZE = LRECL THEN DO;  
    EXCCODE='09007'; SEVERITY='W'; MGMTAREA='PERFORMANCE';  
    EXCDESC1='FIXED LENGTH, ' || PUT(DSORG,$2.) ||  
        ' DATA SET IS UNBLOCKED';  
    EXCDESC2='VOL=' || VOLSER || ', DSN=' || DSNAME;  
    LINK HIT;  
END;
```

THRESHOLD MODIFICATION: No threshold modification is required.

## 09008: DA Data Set is in Multiple Extents

FILE: DAA Data Set Allocation File  
 SAS FILE NAME: DETAIL.VCADAA01  
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Warning (SEVERITY='W')  
 MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies direct data sets (DSORG=DA) whose allocation has expanded beyond a single extent.

RATIONALE: Access to a multiple extent data set is normally less efficient than if the data set were allocated in a single extent because of the overhead involved in positioning to the next data set extent. This is especially true for a direct data set (DSORG=DA) because of the random nature of the access to such a data set. Moving back and forth among the extents is much easier if the data set is accessed randomly rather than sequentially.

DEFINITION: This exception is detected when a data set's organization (DSORG) is direct (DA) and the number of data set extents (DAAEXTNO) is greater than one.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/
/* */
/* 09008 */
/* DSORG=DA DATA SET IS IN MULTIPLE EXTENTS */
/* */
/*-----*/
IF DSORG =: 'DA' AND DAAEXTNO GT ONE THEN DO;
  EXCCODE='09008'; SEVERITY='W'; MGMTAREA='PERFORMANCE';
  EXCDESC1='DSORG=DA DATA SET IS IN MULTIPLE EXTENTS. ' ||
    'EXTENT COUNT=' || PUT(DAAEXTNO,3.);
  EXCDESC2='VOL=' || VOLSER || ', DSN=' || DSNAME;
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: No threshold modification is required.

## 09010: Data Set Using Less than n% of Allocated Space

FILE: DAA Data Set Allocation File  
SAS FILE NAME: DETAIL.VCADAA01  
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)  
  
SEVERITY: Warning (SEVERITY='W')  
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies when a large data set is over-allocated.

RATIONALE: Inefficient use of DASD space can impact the installation's ability to satisfy future DASD space requests. Large data sets that are static in nature should be allocated such that most of the space allocated is being used. If these data sets are over-allocated, the unused space is not available for use.

DEFINITION: This exception is detected when the data set's allocation exceeds the specified value and the percent of allocated space in use falls below the specified minimum.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/  
/* */  
/* 09010 */  
/* DATA SET USING LESS THAN N% OF ALLOCATED SPACE */  
/* */  
/*-----*/  
PERCENT = percent;  
SPACA = spaca;  
IF DAASPACE GE SPACA AND DAAPCPU LT PERCENT THEN DO;  
  EXCCODE='09010'; SEVERITY='W'; MGMTAREA='PERFORMANCE';  
  EXCDISC1='DATA SET USING ONLY' || PUT(DAAPCPU,3.) ||  
    '% OF ALLOCATED SPACE';  
  EXCDISC2='VOL=' || VOLSER || ', DSN=' || DSNAME;  
  LINK HIT;  
END;
```

THRESHOLD MODIFICATION: The user should modify the percent and space values according to the following conventions:

percent - The minimum acceptable value of allocated space

that is in use. A minimum space utilization of 40 percent would be specified as:

```
PERCENT = 40;
```

spaca - The number of allocated tracks that is considered a large allocation. An allocation of 1000 tracks would be specified as:

```
SPACA = 1000;
```

## 09012: VTOC Reorganization Recommended

```
FILE:          VOA Volume Allocation File
SAS FILE NAME:  DETAIL.VCAVOA01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)
```

```
SEVERITY:      Warning      (SEVERITY='W')
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')
```

PURPOSE: Identifies when a VTOC's DSCBs are not localized at the front of the VTOC.

RATIONALE: VTOC search lengths are dependent on a DSCB Format 1 "highwater mark" that is maintained in the Format 4 DSCB. If the used DSCBs have become spread out within the VTOC, the search lengths can increase significantly, which results in poor performance. This spreading effect becomes a problem only when a large amount of VTOC space is in use.

DEFINITION: This exception is based on the number of DSCBs that the Collector scanned to read all of the used DSCBs. This value divided by the number of used DSCBs is a good indication of how spread out the used DSCBs are. A further check is made to see what percent of VTOC DSCBs are in use.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/
/*                                           */
/* 09012                                           */
/* VTOC REORGANIZATION RECOMMENDED           */
/*                                           */
/*-----*/
IF DSCBU AND VOAINXD NE 'Y' THEN DO;
  SPREAD=VOADSCBR/DSCBU;
  PCVTU =DSCBU/VOADSCBS;
  IF SPREAD GE #spread PCVTU GE #pcvtu
  THEN DO;
    EXCCODE='09012'; SEVERITY='W'; MGMTAREA='PERFORMANCE';
    EXCDISC1='VTOC REORGANIZATION RECOMMENDED';
    EXCDISC2='VTOC SPREAD INDEX=' || PUT(SPREAD,3.1) ||
      ', VOL=' || VOLSER;
    LINK HIT;
  END;
END;
```

THRESHOLD MODIFICATION: The user should modify the #spread and #pcvtu values according to the following conventions:

#spread - The maximum acceptable spread index value for DSCBs within the VTOC. A value of 2 means that the usable VTOC is twice as large as it needs to be (i.e., every other DSCB is a free DSCB). A value of 2 would be specified as:

```
IF SPREAD GE 2
```

#pcvtu - The minimum percent of the VTOC in use. Spread indices will not be checked for VTOCs whose percent of space used is below this value. The value is the percent of VTOC DSCBs that are in use. A value of 25 percent would be specified as:

```
IF PCVTU GE 25
```

## 09013: VTOC in Error, Requires Restructuring

FILE: VOA Volume Allocation File  
 SAS FILE NAME: DETAIL.VCAVOA01  
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Impacting (SEVERITY='I')  
 MANAGEMENT AREA: Availability (MGMTAREA='AVAILABILITY')

PURPOSE: Identifies when the DADSM Interrupt Recording Facility (DIRF) indicator is set in the VTOC.

RATIONALE: The DIRF bit is set whenever the DASDSM routines attempt to modify the contents of the VTOC. The DIRF bit is reset when the modification has completed. A system failure during such an update will result in the DIRF bit remaining set.

DEFINITION: This exception is detected when the DIRF bit in the VTOC is set.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/
/*                                          */
/* 09013                                  */
/* VTOC IN ERROR; DIRF BIT SET          */
/*                                          */
/*-----*/
IF VOADIRF = 'Y' THEN DO;
  EXCCODE='09013'; SEVERITY='I'; MGMTAREA='AVAILABILITY';
  EXCDDESC1='VTOC IN ERROR, REQUIRES RESTRUCTURE';
  EXCDDESC2='DIRF BIT SET FOR VOL=' || VOLSER;
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: No threshold modification is required.

## 09014: Tracks Not Accounted for in VTOC

FILE: VOA Volume Allocation File  
SAS FILE NAME: DETAIL.VCAVOA01  
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Impacting (SEVERITY='I')  
MANAGEMENT AREA: Availability (MGMTAREA='AVAILABILITY')

PURPOSE: Identifies those volumes that have tracks not accounted for in the VTOC.

RATIONALE: If due to a system error DASDSM does not successfully complete a function, the VTOC might not be updated correctly. As a result, the VTOC may not account for all of the tracks defined for the device. Tracks unaccounted for cannot be used and are therefore wasted.

DEFINITION: This exception is detected when the number of tracks accounted for in the VTOC does not equal the capacity of the device.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/  
/* */  
/* 09014 */  
/* TRACKS NOT ACCOUNTED FOR IN VTOC */  
/* */  
/*-----*/  
MISSING = CAPAC - (VOASPACA+VOASPACF+VOAVTNTK+ONE);  
IF MISSING THEN DO;  
  EXCCODE='09014'; SEVERITY='I'; MGMTAREA='AVAILABILITY';  
  EXCDESC1='TRACKS NOT ACCOUNTED FOR IN VTOC';  
  EXCDESC2='MISSING TRACK COUNT=' || PUT(MISSING,5.) ||  
    ', VOL=' || VOLSER;  
  LINK HIT;  
END;
```

THRESHOLD MODIFICATION: No threshold modification is required.

## 09015: VTOC Does Not End on a Cylinder Boundary

FILE: VOA Volume Allocation File  
 SAS FILE NAME: DETAIL.VCAVOA01  
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Impacting (SEVERITY='I')  
 MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies when the VTOC's allocation does not end on a cylinder boundary.

RATIONALE: Sequential scanning of the VTOC is sometimes over several tracks. Multiple track searching is handled more efficiently if the data set being scanned is allocated such that the last track of the allocation is the last track in a cylinder.

DEFINITION: This exception is detected when the last track of the VTOC is not the last track of a cylinder.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```

/*-----*/
/*                                          */
/* 09015                                          */
/* VTOC DOES NOT END ON A CYLINDER BOUNDARY      */
/*                                          */
/*-----*/
IF VOAVTNTK GT ONE THEN DO;
  VTETK=VOAVTBTK+VOAVTNTK;
  IF MOD(VTETK,VOAEVVSZ) NE ZERO THEN DO;
    VTCB=INT(VOAVTBTK/VOAEVVSZ);
    VTHB=MOD(VOAVTBTK,VOAEVVSZ);
    VTCE=INT(VTETK/VOAEVVSZ);
    VTHE=MOD(VTETK,VOAEVVSZ);
    EXCCODE='09015'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
    EXCDDESC1='VTOC DOES NOT END ON A CYLINDER BOUNDARY';
    EXCDDESC2='VTOC EXTENT (CCCHH-CCCHH)=' ||
              PUT(VTCB,Z3.) || PUT(VTHB,Z2.) || '-' ||
              PUT(VTCE,Z3.) || PUT(VTHE,Z2.) ||
              ', TRK/CYL=' || PUT(VOAEVVSZ,Z2.) ||
              ', VOL=' || VOLSER;
    LINK HIT;
  END;
END;

```

THRESHOLD MODIFICATION: No threshold modification is required.

## 09016: Large Data Set Not Cylinder Allocated

FILE: DAA Data Set Allocation File  
SAS FILE NAME: DETAIL.VCADAA01  
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Warning (SEVERITY='W')  
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies large data sets that are not cylinder allocated.

RATIONALE: Because of the implementation of data security on DASD devices, overhead increases significantly when multiple-track data processing is performed on data sets that are track-allocated instead of cylinder-allocated. For track-allocated data sets, the system must interrupt processing each time an attempt is made to access a different track. A check is made to see if the new track request falls within the bounds of the data set. For cylinder-allocated data sets this check is performed only when an attempt is made to access a different cylinder. If the data set being accessed is large and is track allocated, this overhead value can become significant.

DEFINITION: The exception is detected when a data set's allocation exceeds the specified value and the type of allocation for the data set is not cylinder.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```

/*-----*/
/*                                           */
/* 09016                                           */
/* LARGE DATA SET NOT CYLINDER ALLOCATED        */
/*                                           */
/*-----*/
SPACA = spaca;
IF DAASPACE GE SPACA AND DAAALTY NE 'CYL' THEN DO;
  EXCCODE='09016'; SEVERITY='W'; MGMTAREA='PERFORMANCE';
  EXCDISC1='LARGE DATA SET NOT CYLINDER ALLOCATED';
  EXCDISC2='VOL=' || VOLSER || ', DSN=' || DSNAME;
  LINK HIT;
END;

```

THRESHOLD MODIFICATION: The user should modify the spaca value according to the following convention:

spaca - The number of allocated tracks that is considered a large allocation. An allocation of 1000 tracks would be specified as:

```
SPACA = 1000;
```

## 09017: VTOC is Not in Indexed Format

```

FILE:          VOA Volume Allocation File
SAS FILE NAME:  DETAIL.VCAVOA01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

```

```

SEVERITY:      Impacting      (SEVERITY='I')
MANAGEMENT AREA: Performance  (MGMTAREA='PERFORMANCE')

```

PURPOSE: Identifies when the VTOC contains an Indexed VTOC data set but is not using it.

RATIONALE: Use of Indexed VTOCs improves system performance by reducing sequential searches of the VTOC and reducing the number of I/Os necessary to maintain the VTOC. If a volume has an Indexed VTOC located on it but the VTOC indicates that normal VTOC processing is being used, then for some reason the VTOC was converted back to normal OS format and the benefits of using an Indexed VTOC are lost.

**DEFINITION:** This exception is detected when an Indexed VTOC data set is allocated on the volume but the Indexed VTOC indicator is not set.

**EXCEPTION STATEMENTS:** The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/
/*                                           */
/* 09017                                           */
/* VTOC IS NOT IN INDEXED FORMAT                                           */
/*                                           */
/*-----*/
IF VOAIXDS = 'Y' AND VOAINDXD = 'N' THEN DO;
  EXCCODE='09017'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDDESC1='VTOC IS NOT IN INDEXED FORMAT';
  EXCDDESC2='BUT VOL=' || VOLSER ||
    ' CONTAINS A ''SYS1.VTOCIX.'' DATA SET.';
  LINK HIT;
END;
```

**THRESHOLD MODIFICATION:** No threshold modification is required.

## 09018: Indexed VTOC Potentially in Error

FILE: VOA Volume Allocation File  
 SAS FILE NAME: DETAIL.VCAVOA01  
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Critical (SEVERITY='C')  
 MANAGEMENT AREA: Reliability (MGMTAREA='RELIABILITY')

PURPOSE: Identifies that a VTOC is using an Indexed VTOC but the DOS bit is not set.

RATIONALE: Use of an Indexed VTOC is indicated in the VTOC's Format 4 DSCB. Because an Indexed VTOC changes the format of a standard VTOC, the DOS convert bit is also set to indicate that the normal free space computations usually found in the Format 5 DSCBs are no longer correct. If Indexed VTOC support is not available on all systems in a multiple-system complex, a system that does not have the support may attempt to convert a VTOC because the DOS bit was set by another system that has converted the VTOC to use Indexed VTOC support. This can result in the VTOC being incorrectly updated.

DEFINITION: This exception is detected when an Indexed VTOC data set is being used but the DOS bit has not been set.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/
/*                                          */
/* 09018                                  */
/* INDEXED VTOC POTENTIALLY IN ERROR      */
/*                                          */
/*-----*/
IF VOAVTOCI = '0.....1'B THEN DO;
  EXCCODE='09018'; SEVERITY='C'; MGMTAREA='RELIABILITY';
  EXCDISC1='INDEXED VTOC POTENTIALLY IN ERROR.';
  EXCDISC2='INDEX VTOC INDICATOR SET BUT DOS BIT NOT SET.' ||
    ', VOL=' || VOLSER;
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: No threshold modification is required.

## 09019: Data Set Has An Invalid DS1LSTAR Value

FILE: DAA Data Set Allocation File  
SAS FILE NAME: DETAIL.VCADAA01  
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYVCAEXC)

SEVERITY: Warning (SEVERITY='W')  
MANAGEMENT AREA: Performance (MGMTAREA='AVAILABILITY')

PURPOSE: Identifies when the data set has an invalid value for DS1LSTAR (last track and block used in TTR format).

RATIONALE: The value for DS1LSTAR is used to set the value for space used in tracks (DAASPACU) for non-VSAM. This value sometimes can be corrupted if errors occur during CLOSE processing. Data sets with an invalid DS1LSTAR should be investigated and repaired if possible with the appropriate utility function (COPY, COMPRESS, EDIT(SAVE), etc.).

DEFINITION: This exception is detected when the value for DAALSTAR (read in from format 1 DSCB field called DS1LSTAR) exceeds its maximum valid value of x'FFFFFF' or 16777215.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
/*-----*/  
/* 09019 */  
/* INVALID DS1LSTAR VALUE */  
/*-----*/
```

```
IF DAALSTAR GE 16777215
```

```
THEN DO;  
EXCCODE = '09019'; SEVERITY = 'W'; MGMTAREA =  
  'AVAILABILITY';  
EXCDESC1 = 'INVALID DS1LSTAR VALUE';  
EXCDESC2 = 'VOL=' || VOLSER || ', DSN=' || DSNAME;  
LINK HIT;  
END;
```

THRESHOLD MODIFICATION: None is suggested.

# Chapter 5: FILES

---

The CA MICS Space Analyzer divides its data logically into the VTOC/Catalog Activity Information Area (VCA), the DFHSM Inventory Primary Information Area (HSM), and the DFHSM Inventory Backup Information Area (HSB). Primary DASD volume and data set information (both VSAM and non-VSAM) comprises the VTOC/Catalog Activity Information Area data files. HSM Level 1 and Level 2 migration information is reflected in the DFHSM Inventory Primary Information Area files. Finally, Hierarchical Storage Backup activity is in the DFHSM Inventory Backup Information Area data files. Figure 5-1 lists the product's files and the timespans that are activated for each file.

The table below identifies the individual database files and categorizes them by the information area to which they belong. For each file in an information area the following data is provided:

**XDWMYT** - This entry defines the timespans in which the file is supported. A nonblank indicates that the file is supported. A period (.) indicates that the file is not supported. The timespans indicated by XDWMYT are as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA

**File** - The name use to access this file.

**File Name** - The descriptive label for the file.

**Note:** The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

USS File System		Date Generated:
Information Area		Tue, May 12, 2009
XDWMYT	File	File Name
XD.M..	HFSDIR	USS DIRECTORY ENTRY FILE
VTOC/Catalog Activity		Date Generated:
Information Area		Tue, May 12, 2009
XDWMYT	File	File Name
XD.M..	VCA_VS	VSAM DATA SET ALLOCATION FILE
XD.M..	VCADAA	DATA SET ALLOCATION FILE
.D.M..	VCADCS	DATA CLASS SUMMARY FILE
X..M..	VCAVOA	VOLUME ALLOCATION FILE
DFHSM Inventory Backup		Date Generated:
Information Area		Tue, May 12, 2009
XDWMYT	File	File Name
XD.M..	HSBBAC	BACKUP DATA FILE
DFHSM Inventory Primary		Date Generated:
Information Area		Tue, May 12, 2009
XDWMYT	File	File Name
XD.M..	HSMMIG	MIGRATED DATA FILE
XD.WMY.	HSMTAP	TAPE PLANNING FILE
XD.MY.	HSMVOL	DASD VOLUME PLANNING FILE

Figure 5-1. HSF, VCA, HSM, and HSB Files

This section contains the following topics:

[5.1 Data Element Naming Conventions](#) (see page 105)

[5.2 VTOC/Catalog Activity Information Area Files](#) (see page 106)

[5.3 DFHSM Inventory Primary Information Area Files](#) (see page 148)

[5.4 DFHSM Inventory Backup Information Area Files](#) (see page 170)

[5.5 USS File System Information Area File](#) (see page 179)

## 5.1 Data Element Naming Conventions

CA MICS data elements follow naming conventions that depend on whether they are standard or common data elements. Standard data elements use the first three characters of their name to identify the file in which they are defined. The following charts list the three-character prefixes with which the standard data element names begin and the files in which they are contained.

### VCA Information Area

File Name	File	Names Begin With
Data Set Allocation	VCADAA	DAA
VSAM Data Set Allocation	VCA_VS	DAA (see Note)
Volume Allocation	VCAVOA	VOA
Data Class Summary	VCADCS	DAA (see Note)

### DFHSM Inventory Primary Information Area

File Name	File	Names Begin With
Migrated Data	HSM MIG	MIG
Volume Planning	HSM VOL	VOL
Tape Planning	HSM TAP	TAP

DFHSM Inventory Backup Information Area

File Name	File	Names Begin With
BACKUP Data	HSBBAC	BAC

NOTE: The VCA\_VS and VCADCS files are called parallel files because they contain the same elements as another file. For these files, the first three characters of the data element names are not the same as the last three characters of the file name.

Common data elements do not use a data element prefix. They have a common definition across database information areas or across files within an information area. You will find common data elements listed under Sequence/Summary Data Elements and Common Data Elements in the Data Elements List sections that follow.

## 5.2 VTOC/Catalog Activity Information Area Files

This section identifies each file in the VTOC/Catalog Activity Information Area and defines its levels of summarization and data sequencing. Also, it presents the list of available data elements for each file.

The files maintained in the VTOC/Catalog Activity Information Area include:

- 1 - Data Set Allocation File (VCADAA)
- 2 - VSAM Data Set Allocation File (VCA\_VS)
- 3 - Volume Allocation File (VCAVOA)
- 4 - Data Class Summary File (VCADCS)

## 5.2.1 Data Set Allocation File (VCADAA)

FILENAME=VCADAA

The Data Set Allocation File describes the attributes, allocation, and current status for each data set in the installation.

This file is derived from data collected by the DASD Space Collector (VCC). One observation is built for each data set on each volume selected for processing. If a data set resides on more than one volume, then that data set will have more than one observation in the file.

Sources of data for the observations in this file are as follows:

1. The format-1 DSCB in the VTOC provides the data set name and most of the attributes of the data set.
2. The SMS cells in the VVDS provide the SMS class name values for DATACLAS, STORCLAS, and MGMTCLAS constructs if the data set is SMS-managed.
3. The storage group name is obtained by VCC from the SMS address space if available.

The VCA DAA file contains observations for both non-VSAM and VSAM data sets. For non-VSAM observations, the VCADAA file is sole location of data, having been merged into a complete observation from the above sources.

For VSAM observations, it is important to realize that the data for the VCADAA file comes from the VTOC, not the catalog. With ICF catalogs, the VTOC is the definitive source of information for the space allocated by a data set (DAASPACA). For this reason, the VCADAA file is perfectly adequate for accounting applications for both VSAM and non-VSAM. The utilization of space within an allocated extent is not described in the VTOC for VSAM files; hence, when the data set type (DAADSTYP) is VS (for VSAM), space used (DAASPACU) is always set equal to space allocated (DAASPACA). Yet, for accounting purposes, space allocated and its time-oriented derivatives such as DAADURSP and DAAKBHR are always sufficient.

Another reason why VSAM data is often better observed through the VCADAA file is the maintenance of various fields by MVS and DFP. For example, the critically important element known

as date last-referenced (DAAREFDT) is more accurate when obtained from the VTOC field known as DSIREFD.

Yet another reason why the VCADAA file should be considered as a source of data about VSAM data sets is simply ease of programming. It is generally easier to report on data from one SAS file if you are new to SAS programming. Unless you need detailed attributes unique to VSAM structures (number of control-area splits (DAANCAS)), or you need utilization statistics within the allocated space, you should consider using the VCADAA file for all data set types due to the simplicity of this approach.

The following sections describe the file's organization, list the data elements maintained, and provide usage hints.

- 1 - VCADAA File Organization
- 2 - VCADAA Data Elements List
- 3 - VCADAA Usage Considerations
- 4 - VCADAA Retrieval Examples

### 5.2.1.1 VCADAA File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	VCAACT1	VCAACT2	VCAACT3	DAADSTYP	
	DEVTYPE	STORNAME	STORSTGC	STORMGTC	YEAR	
	MONTH	DAY	HOUR	DSNAME	ENDTS	
DAYS	SYSID	VCAACT1	VCAACT2	VCAACT3	DAADSTYP	
	DEVTYPE	STORNAME	STORSTGC	STORMGTC	YEAR	
	MONTH	DAY	HOUR			
WEEKS	N/A					
MONTHS	SYSID	VCAACT1	VCAACT2	VCAACT3	DAADSTYP	
	DEVTYPE	STORNAME	STORSTGC	STORMGTC	YEAR	
	MONTH					
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-2. VCADAA Time-Span Granularity Chart

### 5.2.1.2 VCADAA Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

**TIMESPAN:** Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

**DATA ELEMENT:** The data element name.

**DATA ELEMENT DESCRIPTION:** The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

**GENERATION DATE:** Tue, Jan 15, 2013

**Note:** Essential data elements are identified by an "E" under the Timespan asterisk (\*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

#### Sequence/Summary Data Elements

XD.M..E	DAADSTYP	- Data Set Type
XD...E	DAY	- Day of Month
XD.M..E	DEVTYPE	- Device Type
X....E	DSNAME	- Data Set Name
XD...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year

XD.M..E STORMGTC - DFSMS Management Class Name  
XD.M..E STORNAME - DFSMS Storage Group Name  
XD.M..E STORSTGC - DFSMS Storage Class Name  
XD.M..E SYSID - System Identifier  
XD.M..E VCAACT1 - DIVISION  
XD.M..E VCAACT2 - PROJECT  
XD.M..E VCAACT3 - USER  
XD....E WEEK - Week of Year  
XD.M..E YEAR - Year of Century

Common Data Elements

X.....E BLKSIZE - Data Set Physical Block Size  
X.....E CREATEDT - Creation Date  
XD....E DAYNAME - Name of Day of Week  
X.....E DEVADDR - Device Address  
X.....E DEVNUM - Device Number  
X.....E DSORG - Data Set Organization  
XD.M..E DURATION - Recording Interval Time  
XD.M..E ENDTS - End Time Stamp  
X.....E EXPDT - EXPIRATION DATE  
X.....E LRECL - Data Set Logical Record Length  
X.....E ORGSYSID - Originating System Identification  
X.....E RECFM - Data Set Record Format  
X.....E STORDATC - DFSMS Data Class Name  
XD.M..E VCALVFLG - Large Volume Flag  
X.....E VOLSER - Volume Serial Number

Retained Data Elements

X.....E DAAALTYP - Allocation Type  
X.....E DAAARPN - PDSE High Alloc. Relative Page No.  
X.....E DAAVGRD - Average Record Allocation Units  
X.....E DAACSTAT - Catalog Status (Non-VSAM only)  
X.....E DAAEXPDT - Data Set Expiration Date (Char. Value)  
X.....E DAAKEYL - Key length  
X.....E DAALBKTS - Managed Data Set Last Backup Time Stamp  
X.....E DAALSTAR - TTTR of End of Data  
X.....E DAAMEMCT - PDSE Members Count  
X.....E DAAOPTCD - Access Method Optional Services  
X.....E DAAPIUCT - PDSE Pages in Use Count  
X.....E DAARECNO - Data Set Record Segment Number  
X.....E DAAREFDT - Data Set Last Reference Date  
X.....E DAARKP - Relative Key Position  
X.....E DAASCALO - Secondary Allocation  
X.....E DAASECUR - Data Set Security  
X.....E DAAUPDTE - Data Set Update Indicator  
X.....E DAAURPN - PDSE High Used Relative Page No.  
X.....E DAAVALTY - VSAM Type of Space Allocation

X.....E DAAVLSEQ - Volume Sequence Number  
X.....E DAAXSZ01 - Data Set Extent 01 Track Size  
X.....E DAAXTK01 - Data Set Extent 01 Track Start

## Accumulated Data Elements

XD.M..E DAAALLSP - Amount of Space Allocated (MBytes)  
XD.M..E DAACCU - Device Attached to CU Flag  
XD.M..E DAACDSIZ - Compressed Data Set Size (Bytes)  
XD.M..E DAACDSKB - Compressed Data Set Size (KBytes)  
XD.M..E DAACDSMB - Compressed Data Set Size (MBytes)  
XD.M..E DAACMPDS - Number of Compressed Data Sets  
XD.M..E DAACMSAE - Number of CMS Extents Allocated  
XD.M..E DAACMSAT - Amount of CMS Space Allocated (Tracks)  
XD.M..E DAACMSE - Number of Data Sets With CMS Extents  
XD.M..E DAACOST - Space Charges  
XD.M..E DAADSNNO - Number of Data Sets  
XD.M..E DAADUALC - Device Uses Dual Copy Flag  
XD.M..E DAADURSP - Amount of Space-Time (MByte-Hours)  
XD.M..E DAAEADS - Number of Extended Address Data Sets  
XD.M..E DAAEASE - Number of Data Sets EAS Eligible  
XD.M..E DAAEAV - Number of Data Sets Allocated on EAV  
XD.M..E DAAEFDS - Number of Extended Format Data Sets  
XD.M..E DAAEXTNO - Number of Extents Allocated  
XD.M..E DAAHFS - Number of HFS Data Sets  
XD.M..E DAAKBHR - Amount of Space-Time (KBytes-Hours)  
XD.M..E DAAKBUSN - Amount of Space Used (KBytes) - VSAM  
XD.M..E DAAKBYTA - Amount of Space Allocated (KBytes)  
XD.M..E DAAKBYTU - Amount of Space Used (KBytes)  
XD.M..E DAALFDS - Number of Large Format Data Sets  
XD.M..E DAAPDPLX - Primary Duplex Device Flag  
XD.M..E DAAPDSE - Number of PDSE Data Sets  
XD.M..E DAARAIID - Device Uses RAID Technology Flag  
XD.M.. DAAREBLK - Number of Reblockable Data Sets  
XD.M..E DAARLSI - Number of RLS DSNs in Use  
XD.M..E DAARLSQ - Number of RLS DSNs VSAM Quiesced  
XD.M..E DAARLSRR - Number of RLS DSNs Required Recovery  
.D.M..E DAASAMPL - Count of VCC Sample Runs  
XD.M..E DAASDPLX - Secondary Duplex Device Flag  
XD.M..E DAASMSID - Number of System Managed Data Sets  
XD.M..E DAASPACA - Amount of Space Allocated (Tracks)  
XD.M..E DAASPACU - Amount of Space Used (Tracks)  
XD.M..E DAASPAUN - Amount of Space Used (Tracks) - VSAM  
XD.M..E DAATKHR - Amount of Space-Time (Track-Hours)  
XD.M..E DAATMSAE - Number of TMS Extents Allocated  
XD.M..E DAATMSAT - Amount of TMS Space Allocated (Tracks)  
XD.M..E DAATMSE - Number of Data Sets With TMS Extents  
.D.M..E DAATTBLK - Sum of Block (BLKSIZE) Lengths  
.D.M..E DAATTREC - Sum of Record (LRECL) Lengths

XD.M..E DAAUDSIZ - Uncompressed Data Set Size (Bytes)  
XD.M..E DAAUDSKB - Uncompressed Data Set Size (KBytes)  
XD.M..E DAAUDSMB - Uncompressed Data Set Size (MBytes)  
XD.M..E DAAUNCAT - Number of Uncataloged Managed Data Sets  
XD.M..E DAAUSESP - Amount of Space Used (MBytes)  
XD.M..E DAAUSSPN - Amount of Space Used (MBytes) - VSAM  
XD.M..E DAAZFS - Number of ZFS Data Sets

### Maximum Data Elements

.D.M..E DAAMPSPU - Max Percentage of Space Used  
.D.M..E DAAMPSUN - Max Percentage of Space Used - VSAM  
.D.M..E DAAMXEXA - Max Number of Extents Allocated  
.D.M..E DAAMXSPA - Max Amount of Space Allocated (Tracks)  
.D.M..E DAAMXSPU - Max Amount of Space Used (Tracks)  
.D.M..E DAAMXSUN - Max Amount of Space Used (Tracks) - VSAM

### Derived Data Elements

.D.M..E DAAAVBLK - Average BLKSIZE  
.D.M..E DAAAVREC - Average LRECL  
X.....E DAAPCFPU - Pct PDSE Pages Formatted For Use  
X.....E DAAPCIU - Pct PDSE Pages Used  
XD.M..E DAAPCPU - Percentage of Space Used  
XD.M..E DAAPCSUN - Percentage of Space Used - VSAM

### 5.2.1.3 VCADAA Usage Considerations

This section identifies any special considerations or techniques related to using the VCADAA file.

#### Relationship to Data Sources from VCC

VCADAA is built from the data in the VTOC. As such, it contains observations for all data sets on scanned primary storage including VSAM and any DFHSM ML1 data sets (present only if ML1 volumes are not EXCLUDED).

While it contains observations for VSAM data sets, none of this data comes from the catalog structures -- only the VTOC. For details on VSAM data sets, use the VCA\_VS file.

While it may contain observations for data sets under DFHSM management, none of this data comes from the DFHSM MCDS or BCDS -- only the VTOC. For details on inactive data sets or backup versions, use the HSMMIG and HSBAC files respectively.

#### VCADAA Is the Preferred Source For Accounting

We recommend using the VCADAA file when accounting for primary DASD occupancy. The VCADAA file is valid for accounting for both VSAM and non-VSAM. Because the VTOC is the definitive source of what is allocated, it follows that space allocated (DAAALLSP) and its DURATION-oriented derivatives (DAADURSP or DAAKBHR) are always most accurate with the VTOC as a data source.

Note that the above statements only apply to VSAM data sets cataloged in ICF catalogs. Suballocated objects cataloged in pre-ICF catalogs require the VSAM-only view offered by the VCA\_VS file.

Therefore, if you do not have any pre-ICF catalogs still in use (that is, all your catalogs are ICF), then the VCA\_VS file should only be used for accounting if fields in the VCA\_VS file and not in the DAA file are required. Use the VCADAA file instead and consider keeping a few cycles of the VCA\_VS at DETAIL only for VSAM studies of performance.

#### Last-Reference-Date (DAAREFDT) for VSAM and Non-VSAM

VCADAA is the proper source of information for last date referenced (DAAREFDT). While an identically-named variable exists in the VCA\_VS file, only the VCADAA file is accurate because MVS only maintains the date last referenced information in the VTOC.

Beginning with DFP Version 3, the maintenance of last reference date in the VTOC is automatic. If your DFP level is below DFP Version 3, you will need to either install DFHSM or several IBM APARs to capture date of last reference for all access methods.

#### Significance of DAADSTYP Indicator

DAADSTYP identifies the general type of data set:

OS for non-VSAM  
VS for VSAM

HS for a DFHSM-managed object  
HV for a DFHSM-owned VSAM SDSP

Historically, it has been included as a SORT/SEQUENCE variable causing summarization breaks at the DAYS timespan and higher.

Try to avoid using DAADSTYP in your VCAACRT routine since this variable will not be available for the HSMMIG and HSBAC files.

### Observations for VSAM Data Sets

The DETAIL VCADAA file reflects the actual allocated space used by a VSAM Relative Record Data Set (RRDS), Entry Sequenced Data Set (ESDS), Key Sequenced Data Set (KSDS), or Linear Data Set (LDS). Each VSAM data component (and each KSDS key range data component) has an observation for every volume occupied by the VSAM data set. In addition, all KSDS structures have a second set of observations (one per volume) to identify the index component.

Note that there are no observations for CANDIDATE volumes. Candidate volumes are specified when the VSAM data set is defined. They are used only after the data set fills its primary and secondary allocations on the prime volumes. Candidate volumes are no longer supported (needed) in a DFSMS environment.

The first thing that you will note about VSAM data sets in the DETAIL timespan of the VCADAA file is that Space Used (DAASPACU) is equal to Space Allocated (DAASPACA). The second is that there are very few elements describing VSAM data sets.

One other important thing to remember is that only the data set names for the DATA and INDEX components are present; you will not find the cluster name in the VCADAA file. Furthermore, for key range defined VSAM data sets, the DATA and INDEX component names (DSNAME) will be suffixed by a unique four-character qualifier (a letter from A-Z followed by a three-digit sequence number).

### Extra Observations If Greater Than 16 Extents (VSAM)

Because a VSAM data set may have 128 extents and an observation will only hold 16 extents, pseudo observations can be created to represent this additional level of detail about the data set.

Please refer to Section 5.2.2.3 where this is discussed in detail.

#### Multi-Volume Effect On Reporting

To report on the whole VSAM data set you will need to sort (and possibly summarize) the VCADAA file by Data Set Name (DSNAME) followed by Volume Serial Number (VOLSER) to bring the various parts together.

### 5.2.1.4 VCADAA Retrieval Examples

In the examples, a SAS macro variable is used to specify the DDname part of the CA MICS file name. These macro variables are a standard part of CA MICS and are available for all files. The macro variable name has the form &diit, where d is the database identifier, iii is the information area name, and t is the timespan. For the examples, a database identifier of P is used. The identifier is installation dependent, so you should find out what the identifiers are at your installation.

1. List all data sets created in the last 30 days with a high level data set name qualifier of 'CMPR'.

```
DATA;  
SET &PVCAX..VCADAA01;  
IF DSNAME='CMPR';  
IF CREATEDT > DATEPART(ENDTS) - 30;  
PROC PRINT; VAR DSNAME VOLSER CREATEDT DAASPACA DAASPACU;
```

2. Show all non-VSAM data sets having greater than 13 extents and the total megabytes they occupy. Report the largest data sets first.

```
DATA ONE;
  SET &PVCAX..VCADAA01;
  IF DAADSTYP EQ 'OS' AND DAAEXTNO GT 13;
PROC SORT;
  BY DESCENDING DAAALLSP;
PROC PRINT NOOBS UNIFORM N;
  VAR DSNAME DAAEXTNO DEVTYPE VOLSER CREATEDT
      DAAREFDT DAAALLSP;
  SUM DAAALLSP;
  TITLE
  'LISTING OF NON-VSAM DATA SETS WITH MORE THAN 13 EXTENTS';
RUN;
```

3. Show all values of BLKSIZE in the form of a graph. Also show the amount of space allocated by data sets using each BLKSIZE.

```
PROC CHART DATA=&PVCAX..VCADAA01;
  HBAR BLKSIZE / TYPE=SUM SUMVAR=DAAALLSP DISCRETE;
  TITLE1
  'SUMMARY OF MEGABYTES ALLOCATED / BY BLOCK SIZE';
RUN;
```

4. Show the frequency distribution of management class name (MGMTCLAS) using the CA MICS element STORMGTC. Show how management class is distributed within storage group. The element name for storage group is STORNAME.

```
PROC FREQ DATA=&PVCAX..VCADAA01;
  TABLES STORNAME*STORMGTC;
RUN;
```

## 5.2.2 VSAM Data Set Allocation File (VCA\_VS)

The VSAM Data Set Allocation File describes the attributes, allocation, and current status for each VSAM data set in the installation.

This file is derived from data collected by the DASD Space Collector (VCC). One observation is built for each key range for each VSAM object, data or index, on each volume selected for processing. If a VSAM object, data or index, resides on more than one volume, or has more than one key range, then that data set will have more than one observation in the file.

The following sections describe the file's organization, list the data elements maintained, and provide usage hints.

- 1 - VCA\_VS File Organization
- 2 - VCA\_VS Data Elements List
- 3 - VCA\_VS Usage Considerations
- 4 - VCA\_VS Retrieval Examples

### 5.2.2.1 VCA\_VS File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	VCAACT1	VCAACT2	VCAACT3	DAADSTYP	
	DEVTYPE	STORNAME	STORSTGC	STORMGTC	YEAR	
	MONTH	DAY	HOUR	DSNAME	ENDTS	
DAYS	SYSID	VCAACT1	VCAACT2	VCAACT3	DAADSTYP	
	DEVTYPE	STORNAME	STORSTGC	STORMGTC	YEAR	
	MONTH	DAY	HOUR			
WEEKS	N/A					
MONTHS	SYSID	VCAACT1	VCAACT2	VCAACT3	DAADSTYP	
	DEVTYPE	STORNAME	STORSTGC	STORMGTC	YEAR	
	MONTH					
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-3. VCA\_VS Time-Span Granularity Chart

### 5.2.2.2 VCA\_VS Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

**TIMESPAN:** Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

**DATA ELEMENT:** The data element name.

**DATA ELEMENT DESCRIPTION:** The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

**GENERATION DATE:** Tue, Jan 15, 2013

**Note:** Essential data elements are identified by an "E" under the Timespan asterisk (\*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

#### Sequence/Summary Data Elements

XD.M..E	DAADSTYP	- Data Set Type
XD...E	DAY	- Day of Month
XD.M..E	DEVTYPE	- Device Type
X....E	DSNAME	- Data Set Name
XD...E	HOUR	- Hour of Day
XD.M..E	MONTH	- Month of Year

XD.M..E STORMGTC - DFSMS Management Class Name  
XD.M..E STORNAME - DFSMS Storage Group Name  
XD.M..E STORSTGC - DFSMS Storage Class Name  
XD.M..E SYSID - System Identifier  
XD.M..E VCAACT1 - DIVISION  
XD.M..E VCAACT2 - PROJECT  
XD.M..E VCAACT3 - USER  
XD....E WEEK - Week of Year  
XD.M..E YEAR - Year of Century

Common Data Elements

X.....E BLKSIZE - Data Set Physical Block Size  
X.....E CREATEDT - Creation Date  
XD....E DAYNAME - Name of Day of Week  
X.....E DEVADDR - Device Address  
X.....E DEVNUM - Device Number  
X.....E DSORG - Data Set Organization  
XD.M..E DURATION - Recording Interval Time  
XD.M..E ENDTS - End Time Stamp  
X.....E EXPDT - EXPIRATION DATE  
X.....E LRECL - Data Set Logical Record Length  
X.....E ORGSYSID - Originating System Identification  
X.....E RECFM - Data Set Record Format  
X.....E STORDATC - DFSMS Data Class Name  
XD.M..E VCALVFLG - Large Volume Flag  
X.....E VOLSER - Volume Serial Number

Retained Data Elements

X.....E DAAALTYP - Allocation Type  
X.....E DAAASPA - Number of Bytes of Free Space  
X.....E DAAATTRA - AMDSB Data Set Attributes  
X.....E DAAATTR1 - VSAM Data Set Attributes  
X.....E DAAATTR3 - AMDSB Attributes  
X.....E DAABFNO - Number of Data or Index Buffers  
X.....E DAABKTK - Number of Physical Blocks per Track  
X.....E DAABNAME - Alternate Index Base Cluster Name  
X.....E DAABUFSZ - VSAM Minimum Buffer Size  
X.....E DAABYTCA - Number of Bytes per CA  
X.....E DAACATNM - Catalog Name  
X.....E DAACFLAG - VSAM Cluster Flags  
X.....E DAACINV - VSAM Control Interval Size  
X.....E DAACIPCA - Number of CIs in a CA  
X.....E DAACNAME - Cluster Name  
X.....E DAADELRL - Number of Deleted Records  
X.....E DAAEXCP - Number of EXCPs  
X.....E DAAFSCA - Number of Free CIs in a CA  
X.....E DAAFSCI - Number of Free Bytes in a CI

X.....E DAAHARBA - VSAM High Allocated RBA  
 X..... DAAHIKY - High Key on Volume  
 X..... DAAHLRBA - RBA of High Level Index Record  
 X.....E DAAHURBA - VSAM High Used RBA  
 X.....E DAAIMBED - VSAM IMBED Option Indicator  
 X.....E DAAIREC - Number of Inserted Records  
 X.....E DAAKEYL - Key length  
 X.....E DAALBKTS - Managed Data Set Last Backup Time Stamp  
 X..... DAALOWKY - Low Key on Volume  
 X.....E DAAMLRC - Maximum Record Size  
 X.....E DAANCAS - Number of CA Splits  
 X.....E DAANCIS - Number of CI Splits  
 X.....E DAANEST - Number of Index Entries in Index  
 X.....E DAANEXT - Number of Extents in Data Set  
 X.....E DAANIL - Number of Index Levels  
 X.....E DAANLR - Number of User-supplied Records  
 X.....E DAAOWNER - VSAM Data Set Owner  
 X.....E DAAPCTCA - Percentage of Free CIs in the CA  
 X.....E DAAPCTCI - Percentage of Free Bytes in the CI  
 X.....E DAAPFLAG - VCC Processing Flags  
 X.....E DAAPRIMS - Primary Allocation  
 X.....E DAARECNO - Data Set Record Segment Number  
 X.....E DAAREFDT - Data Set Last Reference Date  
 X.....E DAARETR - Number of Retrieved Records  
 X.....E DAARKP - Relative Key Position  
 X.....E DAASCALO - Secondary Allocation  
 X.....E DAASECUR - Data Set Security  
 X.....E DAASHROP - Sharing Options  
 X..... DAASSRBA - RBA of First Sequence Set Record  
 X..... DAASTRNO - Number of Concurrent Requests  
 X..... DAASTSP - OS/VS System Time Stamp  
 X.....E DAATKCA - Number of Tracks per CA  
 X.....E DAAUPR - Number of Updated Records  
 X.....E DAAVCOMP - VSAM Data Set Component Type  
 X.....E DAAXSZ01 - Data Set Extent 01 Track Size  
 X.....E DAAXTK01 - Data Set Extent 01 Track Start

#### Accumulated Data Elements

XD.M..E DAAALLSP - Amount of Space Allocated (MBytes)  
 XD.M..E DAACCU - Device Attached to CU Flag  
 XD.M..E DAACDSIZ - Compressed Data Set Size (Bytes)  
 XD.M..E DAACDSKB - Compressed Data Set Size (KBytes)  
 XD.M..E DAACDSMB - Compressed Data Set Size (MBytes)  
 XD.M..E DAACMPDS - Number of Compressed Data Sets  
 XD.M..E DAACMSAE - Number of CMS Extents Allocated  
 XD.M..E DAACMSAT - Amount of CMS Space Allocated (Tracks)  
 XD.M..E DAACMSE - Number of Data Sets With CMS Extents  
 XD.M..E DAACOST - Space Charges

XD.M..E DAADSNNO - Number of Data Sets  
XD.M..E DAADUALC - Device Uses Dual Copy Flag  
XD.M..E DAADURSP - Amount of Space-Time (MByte-Hours)  
XD.M..E DAAEADS - Number of Extended Address Data Sets  
XD.M..E DAAEASE - Number of Data Sets EAS Eligible  
XD.M..E DAAEAV - Number of Data Sets Allocated on EAV  
XD.M..E DAAEFDS - Number of Extended Format Data Sets  
XD.M..E DAAEXTNO - Number of Extents Allocated  
XD.M..E DAAEXTUS - Number of Extents Used  
XD.M..E DAAKBHR - Amount of Space-Time (KBytes-Hours)  
XD.M..E DAAKBYTA - Amount of Space Allocated (KBytes)  
XD.M..E DAAKBYTU - Amount of Space Used (KBytes)  
XD.M..E DAAPDPLX - Primary Duplex Device Flag  
XD.M..E DAARAIID - Device Uses RAID Technology Flag  
XD.M..E DAARLSI - Number of RLS DSNs in Use  
XD.M..E DAARLSQ - Number of RLS DSNs VSAM Quiesced  
XD.M..E DAARLSRR - Number of RLS DSNs Required Recovery  
.D.M..E DAASAMPL - Count of VCC Sample Runs  
XD.M..E DAASDPLX - Secondary Duplex Device Flag  
XD.M..E DAASMSID - Number of System Managed Data Sets  
XD.M..E DAASPACA - Amount of Space Allocated (Tracks)  
XD.M..E DAASPACU - Amount of Space Used (Tracks)  
XD.M..E DAATKHR - Amount of Space-Time (Track-Hours)  
XD.M..E DAATMSAE - Number of TMS Extents Allocated  
XD.M..E DAATMSAT - Amount of TMS Space Allocated (Tracks)  
XD.M..E DAATMSE - Number of Data Sets With TMS Extents  
.D.M..E DAATTBLK - Sum of Block (BLKSIZE) Lengths  
.D.M..E DAATTREC - Sum of Record (LRECL) Lengths  
XD.M..E DAAUDSIZ - Uncompressed Data Set Size (Bytes)  
XD.M..E DAAUDSKB - Uncompressed Data Set Size (KBytes)  
XD.M..E DAAUDSMB - Uncompressed Data Set Size (MBytes)  
XD.M..E DAAUSESP - Amount of Space Used (MBytes)  
XD.M..E DAAZFS - Number of ZFS Data Sets

Maximum Data Elements

.D.M..E DAAMPSPU - Max Percentage of Space Used  
.D.M..E DAAMXEXA - Max Number of Extents Allocated  
.D.M..E DAAMXEXU - Max Number of Extents Used  
.D.M..E DAAMXSPA - Max Amount of Space Allocated (Tracks)  
.D.M..E DAAMXSPU - Max Amount of Space Used (Tracks)

Derived Data Elements

.D.M..E DAAAVBLK - Average BLKSIZE  
.D.M..E DAAAVREC - Average LRECL  
XD.M..E DAAPCSPU - Percentage of Space Used

### 5.2.2.3 VCA\_VS Usage Considerations

This section identifies special considerations and techniques related to using the VCA\_VS file.

#### Relationship to Catalogs As Data Source

The data for this file is derived from catalog structures. No VTOC data is present in the VCA\_VS file. Actual data sources for the VCA\_VS observations are:

- o ICF catalog structures
  - the BCS component (also known as a user catalog)
  - the VVDS component
- o VSAM catalog structures

If you are still using pre-ICF catalog structures, then suballocated data sets are possible objects unless the VSAM data set was allocated with the UNIQUE keyword. Objects suballocated within such an "old-style" catalog can only be identified by means of the VCA\_VS file.

The DETAIL timespan VCA\_VS file has the same structure as the VCADAA file, but contains detailed information describing a VSAM Relative Record Data Set (RRDS), Entry Sequenced Data Set (ESDS), Key Sequenced Data Set (KSDS), or Linear Data Set (LDS).

#### A Transformation of LISTCAT Data

In many ways, the VCA\_VS file can be viewed as the output of IDCAMS LISTCAT transformed into a SAS file. It is literally an inventory of the catalog structures at a given point in time. There are several SAS data steps and PROC SORTs needed to transform raw VCC data into a cycle of the VCA\_VS file.

#### Effect of EXCLUDEVOL During Data Collection

Excluding volumes from VCC data collection can have an impact on the content of the VCA\_VS file. If a BCS exists on a volume that is not scanned, then the BCS will also not be scanned. If a BCS is not scanned, then only the VVDS component of an ICF catalog structure contributes data toward VCA\_VS observations.

You may specify that VSAM data sets are/are not to be saved in the VCA\_VS file when there are no matching BCS records collected. The BCSREQUIRED option is discussed in Chapter 7.

The default is to accept all VSAM data sets with records from only the VVDS. When this happens, the VCC Processing Flags (DAAPFLAG) are set to indicate the presence or absence of the BCS record as follows:

```
'11.....'B - BCS and VVDS elements are present
'01.....'B - BCS elements are not present
```

We process the BCS to obtain the following data elements. When the BCS records are not present, these data elements are not valid in the VCA\_VS file:

Element	Title
-----	-----
DAAOWNER	VSAM Data Set Owner
DAABNAME	Alternate Index Base Cluster Name
DAACFLAG	VSAM Cluster Flags
DAASECUR	Data Set Security
CREATEDT	Creation Date
EXPDT	Expiration Date

Of these, the most important to you might be the Creation Date (CREATEDT) element. It may be used to compute the DURATION element and could affect some data set charges issued by the CA MICS Accounting and Chargeback product (depending on your accounting journal file options).

#### Extra Observations When Extents Greater Than 16

Because a VSAM data set may have 128 extents and a VCA\_VS observation will (by design) only hold 16 extents, extra (pseudo) observations will be created in both the VCADAA and VCA\_VS files to identify the additional VSAM data set extents. DAARECNO is incremented by 1 for each extra observation required. Consider this simple example:

DSNAME	DAARECNO	DAAEXTNO	DAAXSZ01	DAAXTK01	....
-----	-----	-----	-----	-----	-----
USERID.VSAMTEST.DATA	1	44	15	10025	
USERID.VSAMTEST.DATA	2	0	15	8085	
USERID.VSAMTEST.DATA	3	0	15	12375	

Each extra (pseudo) observation describes the next set of 16 extents. The extra observations take DASD space and time to

build, and are important only if you run the TRACK MAP standard report or do very detailed extent reporting.

All accumulated elements in these pseudo observations (DAASPACA, DAASPACU, DAACOST, etc.) are set to zero and will not affect any summarizations. In the example above, DAAEXTNO indicates 44 extents but that is only shown in the first observation -- the extra observations have had the numeric elements set to zero. Note also, that DAAXSZ01 in the observation with DAARECNO set to 2 is describing the size of extent 17 not extent 1. Likewise, the next observation describes extent 33 under DAAXSZ01 rather than extent 1. This data set is fully described in a total of 3 observations (because 44 is less than 3 times 16). If the data set were in 49 extents, a fourth observation would be written for this data set.

To save processing time and DASD space, you may specify that the extra observations are to be deleted during the DAILY DAY090 step. They may be deleted from either or both of the VCADAA and VCA\_VS files by coding EXTENTDETAIL fff NO and running a VCAPGEN. See Chapter 7 for details.

#### Computation of Space Used For VSAM

While the VCADAA file is the definitive view of space allocated for both non-VSAM and VSAM files, the VCA\_VS file offers the only accurate view of space used. The algorithm used by VCA to compute space used for VSAM data sets is complex. Elements involved in the calculations include:

Element	Title
DAABYTCA	Number of Bytes per CA
DAATKCA	Number of Tracks per CA
DAAHURBA	High-Used Relative Block Address (RBA)
DAACINV	Control Interval Size
DAACIPCA	Number of Control Intervals in a Control Area

Because system page type data sets (PLPA, Common, Page, etc.) can use a special track overflow option not available to other types of VSAM data sets, they may have a fractional number of Control Intervals (CI) on a track (for example, n.3, n.8). When this happens, the method VCA uses to calculate space used is essential for accuracy.

#### Access Method Data Statistics and Attributes

An old-style VSAM Catalog or a VSAM Volume Data Set (VVDS) on

the primary (first used) volume contains the Access Method Data Statistics Block (AMDSB) cell, which records information pertaining to the whole VSAM data set. This set of elements (approximately 40) must be propagated from the first volume and/or first key range's AMDSB cell to all other volume and/or key range observations.

An AMDSB Exception Report is generated on the DAILY step DAY090 MICSL0G output when any AMDSB cell is not present. This can happen when a VSAM catalog or primary VVDS is not processed or is not excluded from collection.

For VSAM data sets allocated on multiple volumes, Key Sequenced (KSDS) VSAM data sets split into multiple key ranges, or KSDSs created with the IMBED option, the following data elements are propagated:

Element	Title
-----	-----
DAAATTR3	AMDSB Attributes
DAAATTR4	AMDSB Data Set Attributes
DAABFNO	Number of Data or Index Buffers
DAACINV	VSAM Control Interval Size
DAACIPCA	Number of CIs in a CA
DAAFSCA	Number of Free CIs in a CA
DAAFSCI	Number of Free Bytes in a CI
DAAHLRBA	RBA of High Level Index Record
DAAKEYL	Key Length
DAAMLRC	Maximum Record Size
DAANEXT	Number of Extents in Data Set
DAAPCTCA	Percentage of Free CIs in the CA
DAAPCTCI	Percentage of Free Bytes in the CI
DAAREFDT	Data Set Last Reference Date
DAARKP	Relative Key Position
DAASSRBA	RBA of First Sequence Set Record
DAASTRNO	Number of Concurrent Requests
DAASTSP	OS/VS System Timestamp
DAAASPA	Number of Bytes of Free Space
DAADELR	Number of Deleted Records
DAAEXCP	Number of EXCPs
DAAIREC	Number of Inserted Records
DAANCAS	Number of CA Splits
DAANCIS	Number of CI Splits
DAANEXT	Number of Extents in Data Set
DAANIL	Number of Index Levels
DAANLR	Number of User-supplied Records
DAARETR	Number of Retrieved Records
DAAUPR	Number of Updated Records

These elements are copied from the Access Method Data Statistics Block (AMDSB) information cell, collected from the VSAM data set's primary volume VVDS, into all observations defining the same VSAM data set. If the AMDSB cell is missing (for example, the primary volume was excluded from the Space Collector run or the volume processing failed), a MICSLOG run time exception report is produced. Any VSAM data sets on this report will contain the above elements set to zero or missing. You should review the VCC run to determine why the AMDSB was not collected.

The following elements are not part of the AMDSB cell but are copied from the primary observation to ensure that all observations for a VSAM data set are consistent:

Element	Title
-----	-----
DAAALTYP	Allocation Type
DAAATTR1	VSAM Data Set Attributes
DAABUFSZ	VSAM Minimum Buffer Size
DAAPRIMS	Primary Allocation
DAASCALO	Secondary Allocation
DAASHROP	Sharing Options
LRECL	Data Set Logical Record Length
RECFM	Data Set Record Format
STORDATC	DFSMS Data Class Name
STORMGTC	DFSMS Management Class Name
STORSTGC	DFSMS Storage Class Name
DAALBKTS	Managed Data Set Last Backup Timestamp
DAASMSID	Number of System Managed Data Sets

#### Key Range Qualifier Suffix in DSNAME

Data set names (DSNAME) for VSAM data sets that have been split into multiple key ranges contain the key range qualifier suffix (.A001, .A002 etc.) for the Data and Index components in the VCA\_VS file.

For key range defined VSAM data sets, the DATA, INDEX, and SSET component names (DSNAME) will be suffixed by a unique four-character qualifier (a letter from A-Z followed by a three-digit sequence number).

#### The IMBED Option and the DAAVCOMP Element

If a KSDS was defined with the IMBED option, there will be a third set of observations (one per data component volume) to define the imbedded sequence set (DAAVCOMP='SSET'). This pseudo observation represents that portion of the data

component used to imbed the low level index record in the first track of each control area. Because it is part of the index, it will have the same data set name as the index component.

#### Multi-Volume Effect On Reporting

To report on a VSAM data set from the VCA\_VS file, you will need to sort (and possibly summarize) by Catalog Name (DAACATNM), Cluster Name (DAACNAME), Component Type (DAAVCOMP), and Data Set Name (DSNAME) followed by Volume Serial Number (VOLSER) to bring the component parts together.

### 5.2.2.4 VCA\_VS Retrieval Examples

In the examples, a SAS macro variable is used to specify the DDname part of the CA MICS file name. These macro variables are a standard part of CA MICS and are available for all files. The macro variable name has the form &diit, where d is the database identifier, iii is the information area name, and t is the timespan. For the examples, a database identifier of P is used. The identifier is installation dependent, so you should find out what the identifiers are at your installation.

1. List all VSAM data sets created yesterday with a high level data set name qualifier of 'PERDB'.

```
DATA;
SET &PVCAX.VCA_VS01;
IF DSNAME='PERDB';
IF TODAY() - 1 = CREATEDT;
PROC PRINT; VAR DSNAME DAACATNM VOLSER
            DAASPACA DAASPACU;
RUN;
```

2. List VSAM structures with 50 or more CA splits or 5000 or more CI splits.

```
DATA;
SET DETAIL.VCA_VS01;
IF DAANCAS GE 50 OR DAANCIS GE 5000;
PROC PRINT NOOBS LABEL;
VAR DSNAME DAAVCOMP VOLSER DAANCAS DAANCIS;
TITLE1 'CI/CA SPLIT REPORT';
```

An example of the output appears as follows:

CI/CA SPLIT REPORT					12:04 FRIDAY, mmmddyyyy
Data Set Name	VSAM Data Set Component Type	Volume Serial Number	Number of CA Splits	Number of CI Splits	
CATALOG.MVSICF1.VMVSXAX	DATA	MVSXAX	44	5819	
CNM.SA08.BNJLGR.DATA	DATA	LRG008	0	14514	
CNM.SA08.AAUVSPL.DATA	DATA	LRG010	116	40623	
CSS.PROD.LMP.SDDS.DATA	DATA	TSU060	140	1401	
CSS.PROD.LMP.SDDS.DATA	DATA	TSU061	140	1401	
CSS.PROD.LMP.SDDS.DATA	DATA	TSU062	140	1401	
CSS.PROD.LMP.SDDS.DATA	DATA	TSU067	140	1401	
CSS.PROD.DB5.SDDS.DATA	DATA	TSU060	150	1408	
CSS.PROD.DB5.SDDS.DATA	DATA	TSU062	150	1408	
KTI004.CED.IPCSDDIR.DATA	DATA	MVS343	32	16400	
ESASMPE.MVSDLIB.CSI.DATA	DATA	S32RS1	33	5332	
ESASMPE.TARGET.CSI.DATA	DATA	S32RS1	32	13112	
INFORAM.TPX.VSAM.DATA	DATA	TSU030	0	13124	
IPOIPCS.RAMJX11.DEBUGD	DATA	MVS965	11	16228	
IPOIPCS.RAMRK11.DEBUGD	DATA	MVS965	55	12736	
IPOIPCS.RAMAM11.DEBUGD	DATA	MVS3C3	64	13653	
IPOIPCS.RAMBP11.DEBUGD	DATA	MVS3C3	9	5412	
IPOIPCS.RAMGM11.DEBUGD	DATA	MVS3C3	25	14727	
IPOIPCS.RAMKB22.DEBUGD	DATA	MVS3C3	26	5101	
IPOIPCS.RAMLP11.DEBUGD	DATA	MVS3C3	43	7941	
IPOIPCS.RAMOS11.DEBUGD	DATA	MVS3C3	10	15829	
IPOIPCS.RAMPL11.DEBUGD	DATA	MVS3C3	31	5276	
IPOIPCS.RAMRC11.DEBUGD	DATA	MVS3C3	51	26015	
IPOIPCS.RAMVK11.DEBUGD	DATA	MVS3C3	7	13391	
IPOIPCS.RAMPR33.DEBUGD	DATA	LRG008	14	14510	
IPOIPCS.RAMAP11.DEBUGD	DATA	TSU028	10	7229	
IPOIPCS.RAMDS11.DEBUGD	DATA	TSU036	6	7264	
IPOIPCS.RAMDV11.DEBUGD	DATA	TSU029	22	6431	
IPOIPCS.RAMG011.DDIRD	DATA	TSU033	42	16766	
IPOIPCS.RAMG011.DEBUGD	DATA	TSU028	28	13393	
IPOIPCS.RAMJK11.DEBUGD	DATA	TSU059	49	5389	
IPOIPCS.RAMJP11.DEBUGD	DATA	TSU027	24	18521	
IPOIPCS.RAMRB11.DEBUGD	DATA	TSU059	24	17467	
IPOIPCS.RAMRF11.DEBUGD	DATA	TSU063	46	6852	
IPOIPCS.RAMTL11.DEBUGD	DATA	TSU027	74	25631	
IPOIPCS.RAMTR11.DEBUGD	DATA	TSU033	24	13514	
IPOIPCS.RAMM11.DEBUGD	DATA	TSU055	17	11823	
SYS9.HSM.BCDS.DATA	DATA	MVS961	87	2294	
RAMPE22.DDIRD	DATA	TSU057	14	6754	
RAMKD11.RAMKD11.DEBUGD	DATA	MVS3C3	27	33061	
TUNDX.REL40.DPMFAIF.DATA	DATA	TSU027	98	716	
TUNDX.REL40.DPMFVPF0.DATA	DATA	TSU027	123	2	
RAMCG22.DDIRD	DATA	TSU027	89	13370	

Figure 5-4. Sample CI/CA Split Report

3. Show the allocated VSAM space in megabytes by storage group.

```
PROC CHART DATA=DAYS.VCA_VS01;
VBAR STORNAME / TYPE=SUM SUMVAR=DAAALLSP;
TITLE1 'ALLOCATED VSAM SPACE (MB) BY STORAGE GROUP';
```

An example of the output appears as follows:

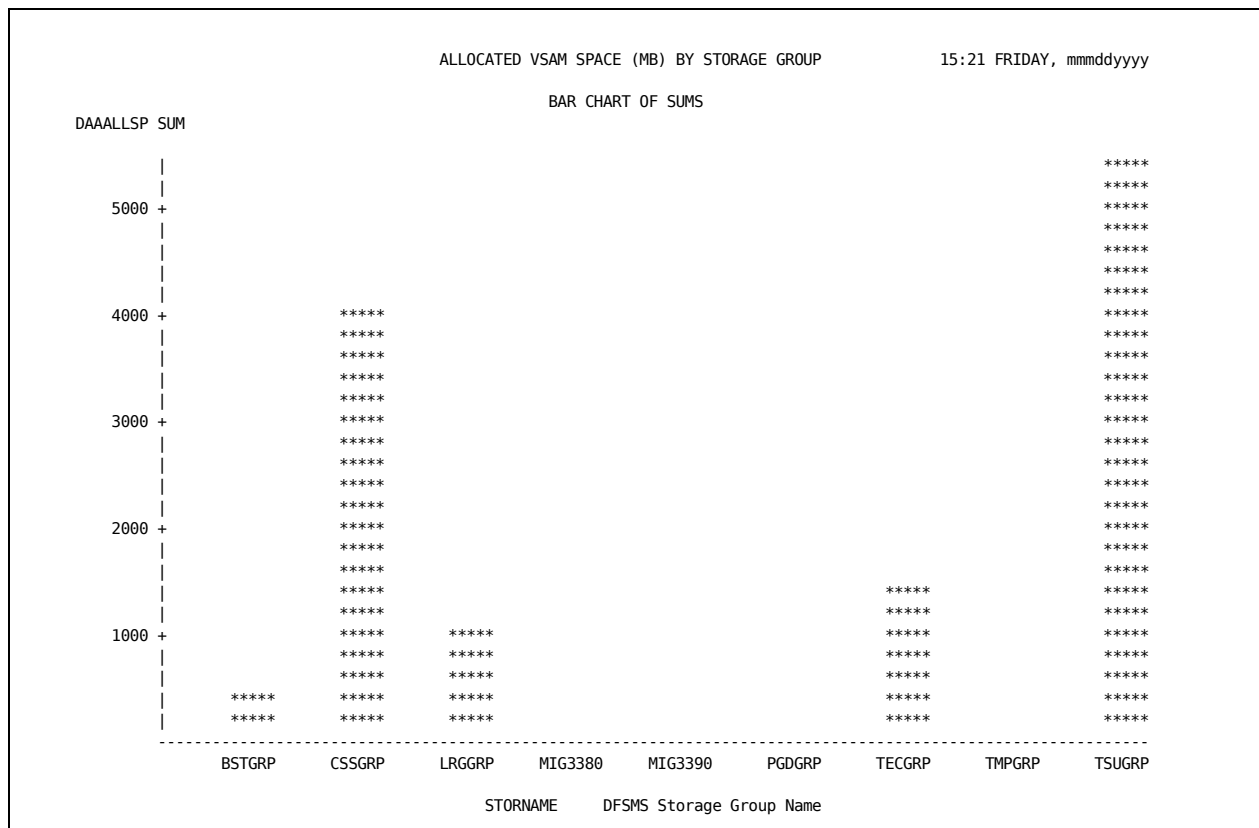


Figure 5-5. Sample Allocated VSAM Space by Storage Group Report

### 5.2.3 Volume Allocation File (VCAVOA)

The Volume Allocation File describes a variety of attributes for each DASD volume scanned by VCC.

Besides the importance of the volume-level statistics kept in the VCAVOA observations, the VCAVOA file is used in a match MERGE operation with both the VCADAA file and the VCA\_VS file in the DAILY run.

The absence of a VCAVOA observation for a data set on a given VOLSER indicates a problem during data collection. VCC accumulates volume-level statistics while reading the VTOC of a volume. When "end-of-VTOC" is reached, a volume level record is written out by VCC. This volume level record (documented in the CA MICS Space Collector Guide) is transformed into a VCAVOA (SAS) observation. Another data step combines calculated VSAM-related totals from WORK versions of the VCA\_VS file to develop the final DETAIL level observation for the VCAVOA file.

If there is not a matching VCAVOA observation for a set of VCADAA observations with a given VOLSER, then all the VCADAA observations for that VOLSER are purged (not written to the VCADAA file). The absence of a VCAVOA observation indicates partial data collection for that scan time (ENDTS) and a report is produced on MICSLOG (see Appendix A - Messages).

The following sections describe the file's organization, list the data elements maintained, and provide usage hints.

- 1 - VCAVOA File Organization
- 2 - VCAVOA Data Elements List
- 3 - VCAVOA Usage Considerations
- 4 - VCAVOA Retrieval Examples

### 5.2.3.1 VCAVOA File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	STORNAME	DEVTYPE	VOLSER	YEAR	
	MONTH	DAY	HOUR	ENDTS		
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	STORNAME	DEVTYPE	VOLSER	YEAR	
	MONTH					
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-6. VCAVOA Time-Span Granularity Chart

### 5.2.3.2 VCAVOA Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

**TIMESPAN:** Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

**DATA ELEMENT:** The data element name.

**DATA ELEMENT DESCRIPTION:** The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

**GENERATION DATE:** Tue, May 12, 2009

**Note:** Essential data elements are identified by an "E" under the Timespan asterisk (\*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

#### Sequence/Summary Data Elements

X....E	DAY	- Day of Month
X..M..E	DEVTYPE	- Device Type
X....E	HOURL	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	STORNAME	- DFSMS Storage Group Name
X..M..E	SYSID	- System Identifier

X..M..E VOLSER - Volume Serial Number  
X....E WEEK - Week of Year  
X..M..E YEAR - Year of Century

Common Data Elements

X....E DAYNAME - Name of Day of Week  
X..M..E DEVADDR - Device Address  
X..M..E DEVNUM - Device Number  
X..M..E DURATION - Recording Interval Time  
X..M..E ENDTS - End Time Stamp  
X..M..E MICSVER - CA MICS Version Number  
X....E ORGSYSID - Originating System Identification  
X..M..E VCALVFLG - Large Volume Flag

Retained Data Elements

X.... VOADATK - Next Alternate Track Address  
X.... VOAMTIM - VSAM Time Stamp  
X....E VOABCS - ICF BCS Data Set on Volume  
X..M..E VOACAPMB - Volume Capacity in Megabytes (Retained)  
X..M..E VOACAPTK - Volume Capacity in Tracks (Retained)  
X..M..E VOACMSFX - CMS Fragmentation Index  
X..M..E VOADEVSZ - Number of Tracks per Cylinder  
X..M..E VOADEVTK - Device Track Length  
X....E VOADIRF - DADSM Interrupt Recording Facility  
X..M..E VOADSCBR - Number of DSCBs Read to Scan VTOC  
X..M..E VOADSCBS - Total Number of DSCBs in VTOC  
X....E VOADSCB0 - Number of Format 0 DSCBs  
X....E VOADSCB2 - Number of Format 2 DSCBs  
X....E VOADSCB3 - Number of Format 3 DSCBs  
X....E VOADSCB5 - Number of Format 5 DSCBs  
X....E VOADSCB9 - Number of Format 9 DSCBs  
X..M..E VOADVNAM - Hardware Device Name  
X..M..E VOADVVEN - Hardware Device Manufacturer  
X..M..E VOAFRAGX - Volume Fragmentation Index  
X..M..E VOAFVIRS - Number of Free VTOC Index Records  
X....E VOAFXL01 - Free Space Extents Threshold Level 1  
X....E VOAFXL02 - Free Space Extents Threshold Level 2  
X....E VOAFXL03 - Free Space Extents Threshold Level 3  
X....E VOAFXL04 - Free Space Extents Threshold Level 4  
X....E VOAFXL05 - Free Space Extents Threshold Level 5  
X....E VOAFXL06 - Free Space Extents Threshold Level 6  
X....E VOAFXL07 - Free Space Extents Threshold Level 7  
X....E VOAFXL08 - Free Space Extents Threshold Level 8  
X..M..E VOAINDXD - Indexed VTOC in Use on Volume  
X..M..E VOAIXDS - Indexed VTOC Data Set on Volume  
X.... VOAMCUV - Allocation Unit for CMS  
X..M..E VOAMOUNT - Volume Mount Attribute

X..M..E VOANOATK - Number of Used Alternate Tracks  
 X..... VOAR2TIM - VSAM Volume/Catalog Match Time Stamp  
 X..M..E VOASGS - DFSMS Storage Group Status  
 X..M..E VOASMSFG - System Managed Volume Status  
 X.....E VOASPF01 - Number of Free Extents in Size 1  
 X.....E VOASPF02 - Number of Free Extents in Size 2  
 X.....E VOASPF03 - Number of Free Extents in Size 3  
 X.....E VOASPF04 - Number of Free Extents in Size 4  
 X.....E VOASPF05 - Number of Free Extents in Size 5  
 X.....E VOASPF06 - Number of Free Extents in Size 6  
 X.....E VOASPF07 - Number of Free Extents in Size 7  
 X.....E VOASPF08 - Number of Free Extents in Size 8  
 X..M..E VOATKCAP - Published Track Capacity in Bytes  
 X..M..E VOATMSFX - TMS Fragmentation Index  
 X..M..E VOATOATK - Number of Total Alternate Tracks  
 X..M..E VOAUSE - Volume Use Attribute  
 X..... VOAVSCRA - Track Address of the VSAM CRA  
 X..... VOAVSIND - VSAM Indicators  
 X..M..E VOAVTBTK - VTOC Begin Track Address  
 X..M..E VOAVTNTK - Number of Tracks in VTOC  
 X.....E VOAVTOCI - VTOC Indicators  
 X..M..E VOAVVDS - ICF VVDS Data Set on Volume

#### Accumulated Data Elements

X..M..E VOALLSP - Amount of Space Allocated (MBytes)  
 X..M..E VOACCU - Device Attached to CU Flag  
 X..M..E VOACMSAT - Amount of CMS Space Allocated (Tracks)  
 X..M..E VOACMSDS - Number of CMS Data Sets  
 X..M..E VOACMSFE - Number of Free CMS Extents on Volume  
 X..M..E VOACMSFT - Amount of CMS Space Free (Tracks)  
 X..M..E VOACMSTT - Amount of CMS Space Total (Tracks)  
 X..M..E VOADSCB1 - Number of Format 1 DSCBs  
 X.....E VOADSCB8 - Number of Format 8 DSCBs  
 X..M..E VOADUALC - Device Uses Dual Copy Flag  
 X..M..E VOAEAV - Number of EAV Volumes With CMS  
 X..M..E VOAEXTFR - Number of Free Extents on Volume  
 X..M..E VOAEXTNO - Number of Extents Allocated on Volume  
 X..M..E VOAEXTUS - Number of Extents Used on Volume  
 X..M..E VOAFRESP - Amount of Space Free (MBytes)  
 X..M..E VOAKAPMB - Volume Capacity in Megabytes (Accum.)  
 X..M..E VOAKAPTK - Volume Capacity in Tracks (Accumulated)  
 X..M..E VOAMISSP - Amount of Space Missing (MBytes)  
 X..M..E VOANVMDS - Number of Managed Non-VSAM Data Sets  
 X..M..E VOANVMSA - Amount of Managed Non-VSAM Space Alloc  
 X..M..E VOAPDPLX - Primary Duplex Device Flag  
 X..M..E VOARAIID - Device Uses RAID Technology Flag  
 X..M..E VOASDPLX - Secondary Duplex Device Flag  
 X..M..E VOASPACA - Amount of Space Allocated (Tracks)

X..M..E VOASPACF - Amount of Space Free (Tracks)  
X..M..E VOASPACM - Amount of Space Missing (Tracks)  
X..M..E VOASPACU - Amount of Space Used (Tracks)  
X..M..E VOASPLDS - Number of Split Data Sets  
X..M..E VOATMSAT - Amount of TMS Space Allocated (Tracks)  
X..M..E VOATMSDS - Number of TMS Data Sets  
X..M..E VOATMSFE - Number of Free TMS Extents on Volume  
X..M..E VOATMSFT - Amount of TMS Space Free (Tracks)  
X..M..E VOATMSTT - Amount of TMS Space Total (Tracks)  
X..M..E VOAUSESP - Amount of Space Used (MBytes)  
X..M..E VOAVDSET - Number of VSAM Data Sets on Volume  
X..M.. VOAVDSPA - Number of VSAM Data Spaces on Volume  
X..M.. VOAVEXNO - Number of VSAM Extents Allocated on Vol  
X..M.. VOAVEXUS - Number of VSAM Extents Used on Volume  
X..M..E VOAVOLNO - Number of Volume Samples  
X..M..E VOAVSMDS - Number of Managed VSAM Data Sets  
X..M..E VOAVSMSA - Amount of Managed VSAM Space Allocated  
X..M..E VOAVSPAA - Amount of VSAM Space Allocated on Vol  
X..M..E VOAVSPAU - Amount of VSAM Space Used on Volume  
X..M.. VOAVSPSA - Amount of VSAM Space Sub-allocated

Minimum Data Elements

X..M..E VOAMNCDT - Min (Oldest) Data Set Creation Date  
X..M..E VOAMNCFX - Min CMS Fragmentation Index  
X..M..E VOAMNCXF - Min CMS Free Space in a Single Extent  
X..M..E VOAMNEXF - Min Free Space in a Single Extent  
X..M..E VOAMNFRG - Min Volume Fragmentation Index  
X..M..E VOAMNTFX - Min TMS Fragmentation Index  
X..M..E VOAMNTXF - Min TMS Free Space in a Single Extent

Maximum Data Elements

X..M..E VOAMPSPU - Max Percentage of Space Used on Volume  
X..M..E VOAMPVSA - Max Percentage Space Allocated to VSAM  
X..M..E VOAMXCFX - Max CMS Fragmentation Index  
X..M..E VOAMXCXF - Max CMS Free Space in a Single Extent  
X..M..E VOAMXEXA - Max Number of Extents Allocated on Vol  
X..M..E VOAMXEXF - Max Free Space in a Single Extent  
X..M..E VOAMXEXU - Max Number of Extents Used on Volume  
X..M..E VOAMXFRG - Max Volume Fragmentation Index  
X..M..E VOAMXRDT - Max Data Set Last Reference Date  
X..M..E VOAMXSPA - Max Amount of Space Allocated on Volume  
X..M..E VOAMXSPF - Max Amount of Free Space on Volume  
X..M..E VOAMXSPU - Max Amount of Space Used on Volume  
X..M..E VOAMXTFX - Max TMS Fragmentation Index  
X..M..E VOAMXTXF - Max TMS Free Space in a Single Extent  
X..M..E VOAMXVSA - Max Amount VSAM Space Allocated on Vol  
X..M..E VOAMXVSU - Max Amount of VSAM Space Used on Vol

#### Derived Data Elements

X..M..E VOAPCSPF - Percentage of Space Free on Volume  
X..M..E VOAPCSPU - Percentage of Allocated Space Used  
X..M..E VOAPCVSA - Percentage of Space Allocated to VSAM

### 5.2.3.3 VCAVOA Usage Considerations

This section identifies special considerations and techniques related to using the VCAVOA file.

#### Storage Group Name (STORNAME)

STORNAME is set to blank unless the actual storage group name is captured by VCC at data collection time. Sites that have implemented volume pooling without DFSMS could use a PROC FORMAT combined with the \_USRSVOA exit point to associate a VOLSER with a poolname and place it in STORNAME.

#### Organization of Extent Information

The observations for each volume in the DETAIL timespan of the Volume Allocation File do not contain variables describing the size and location of each free space extent. Instead, each observation contains an array of eight variables, VOASPF01 through VOASPF08, that are used as free space buckets. Each variable contains the count of free space extents whose lengths fall within the range of space defined for that variable. The range for each variable is:

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

For example, if the value of VOASPF03 is 13, this means that the volume has 13 free extents whose sizes are between 201 and 300 tracks in length.

In sharedprefix.MICS.SOURCE, member \$VCAMSTR defines a SAS macro that assigns the value ranges that determine the above elements. To change these values, you could construct a local modification to this member, carefully preserving sequence numbers and following other procedures discussed in the CA MICS System Modification Guide.

#### Collecting Only Volume Level Statistics

VCC has a run time parameter called DATAINFO=N which causes VCC to skip writing records at the data set level (that is, record segments that would eventually become observations in the VCADAA and VCA\_VS files). The intent of DATAINFO=N collection is to provide an easy means of scanning the VTOCs to obtain various totals (and volume attributes) without incurring the overhead of writing (and then processing into CA MICS) the much more voluminous quantities of detail information about data sets.

Moreover, DATAINFO=N collection suggests the very real feasibility of scheduling VCC volume-level data gathering as frequently as every hour or even every 15 minutes for selected volumes or storage groups for special studies.

With DATAINFO=N data collection, the following elements are set to missing values because they are only attainable through the summarization of data from the VCA\_VS file:

VOAVSPSA VSAM Space Allocated  
VOAVSPAU VSAM Space Used  
VOAVEXNO VSAM Extents Allocated  
VOAVEXUS VSAM Extents Used  
VOAVDSET VSAM Data Set Count

Various tradeoffs make this acceptable:

- o More important levels of allocation such as VOASPACA and capacity such as VOACAPMB are accurate and always available.
- o Volume-level only data collection by means of DATAINFO=N should be processed in a separate database unit -- distinctly different in configuration and options from the production unit that contains data set level files (VCADAA,VCA\_VS,HSMMIG,HSBBAC) and Accounting JOURNAL files.
- o The audience for this data is probably confined to

storage administrators and data center management.

#### Averages from Summarized Data Using VOAVOLNO

Determining the average value of a numeric (accumulated) data element requires division by a special count data element called VOAVOLNO.

For example, VOASPACA and other numeric elements are accumulated. Therefore, to show the average space allocated for a month, VOASPACA must be divided by VOAVOLNO. VOAVOLNO is a count of the number of times VCC scanned the volume during the month.

In contrast, VOACAPMB (capacity of a volume in megabytes), is a retained data element and treated as a constant. Therefore it's value at the DETAIL timespan is propagated into the higher timespans during summarization. The capacity of a VOLSER with a given DEVTYPE is assumed to be constant over time.

This same principle holds true at the DAYS timespan if VCC takes an inventory of the volume more than once per day.

### 5.2.3.4 VCAVOA Retrieval Examples

In the examples, a SAS macro variable is used to specify the DDname part of the CA MICS file name. These macro variables are a standard part of CA MICS and are available for all files. The macro variable name has the form &diit, where d is the database identifier, ii is the information area name, and t is the timespan. For the examples, a database identifier of P is used. The identifier is installation dependent, so you should find out what the identifiers are at your installation.

1. List all volumes that exceeded 90% of space used last month, showing the maximum amount of space allocated and used.

```
DATA;  
SET &PVCAM.VCAVOA01;  
IF VOAMPSPU > 90;  
PROC PRINT; VAR VOLSER STORNAME DEVTYPE VOAMPSPU VOAMXSPA  
              VOAMXSPU;  
RUN;
```

2. Show the total megabytes of capacity available and allocated space in megabytes for each volume with control breaks and totals by storage group.

```
DATA WORK.SUBSET  
(KEEP=VOLSER VOANOATK VOAFVIRS VOACAPMB  
  VOASPACA VOATKCAP STORNAME  
  VOAALLMB);  
SET &PVCAX.VCAVOA01;  
VOAALLMB = (VOASPACA*VOATKCAP)/1000000;  
RUN;  
PROC SORT DATA=WORK.SUBSET;  
  BY STORNAME VOLSER;  
RUN;  
PROC PRINT LABEL DATA=WORK.SUBSET;  
  BY STORNAME;  
  SUM VOANOATK VOAFVIRS VOACAPMB VOAALLMB;  
  VAR STORNAME VOLSER VOACAPMB VOAALLMB;  
  SUMBY STORNAME;  
RUN;
```

Note that VOASPACA (carried in tracks) is converted to bytes by multiplying by VOATKCAP (the capacity of the track in bytes). We divide by one million since that is the basis for the published capacity of volume geometries that is carried

as VOACAPMB. A temporary derived element for allocated space in terms of megabytes is created as VOAALLMB.

### 5.2.4 Data Class Summary File (VCADCS)

The Data Class Summary File is only of interest to sites running MVS/SP Version 3 and above. It shows a summarized view of the DETAIL data in the VCADAA and VCA\_VS files with an emphasis on data class (STORDATC or as known in IBM terminology, DATACLAS). Data class is a name that can be used to represent the physical attributes of a data set. Coding or defaulting to a given data class name enables the end-user to reduce the JCL required for allocation of new data sets.

The following sections describe the file's organization, list the data elements maintained, and provide usage hints.

- 1 - VCADCS File Organization
- 2 - VCADCS Data Elements List
- 3 - VCADCS Usage Considerations
- 4 - VCADCS Retrieval Examples

### 5.2.4.1 VCADCS File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	N/A					
DAYS	SYSID	VCAACT1	VCAACT2	VCAACT3	STORDATC	
	YEAR	MONTH	DAY	HOURL		
WEEKS	N/A					
MONTHS	SYSID	VCAACT1	VCAACT2	VCAACT3	STORDATC	
	YEAR	MONTH				
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-7. VCADCS Time-Span Granularity Chart

### 5.2.4.2 VCADCS Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

**TIMESPAN:** Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL  
 D - DAYS  
 W - WEEKS  
 M - MONTHS  
 Y - YEARS  
 T - TABLES AREA  
 . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

**DATA ELEMENT:** The data element name.

**DATA ELEMENT DESCRIPTION:** The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

**GENERATION DATE:** Tue, May 12, 2009

**Note:** Essential data elements are identified by an "E" under the Timespan asterisk (\*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

#### Sequence/Summary Data Elements

.D...E	DAY	- Day of Month
.D...E	HOURL	- Hour of Day
.D.M..E	MONTH	- Month of Year
.D.M..E	STORDATC	- DFSMS Data Class Name
.D.M..E	SYSID	- System Identifier
.D.M..E	VCAACT1	- DIVISION
.D.M..E	VCAACT2	- PROJECT

.D.M..E VCAACT3 - USER  
.D....E WEEK - Week of Year  
.D.M..E YEAR - Year of Century

Common Data Elements

.D....E DAYNAME - Name of Day of Week  
.D.M..E DURATION - Recording Interval Time  
.D.M..E ENDTS - End Time Stamp  
.D.M..E MICSVER - CA MICS Version Number  
.....E ORGSYSID - Originating System Identification

Accumulated Data Elements

.D.M..E DAAALLSP - Amount of Space Allocated (MBytes)  
.D.M..E DAACOST - Space Charges  
.D.M..E DAADSNO - Number of Data Sets  
.D.M..E DAADURSP - Amount of Space-Time (MByte-Hours)  
.D.M.. DAAREBLK - Number of Reblockable Data Sets  
.D.M..E DAASAMPL - Count of VCC Sample Runs  
.D.M..E DAASMSID - Number of System Managed Data Sets  
.D.M..E DAATTBLK - Sum of Block (BLKSIZE) Lengths  
.D.M..E DAATTREC - Sum of Record (LRECL) Lengths  
.D.M..E DAAUNCAT - Number of Uncataloged Managed Data Sets  
.D.M..E DAAUSESP - Amount of Space Used (MBytes)

Derived Data Elements

.D.M..E DAAVBLK - Average BLKSIZE  
.D.M..E DAAVREC - Average LRECL

### 5.2.4.3 VCADCS Usage Considerations

This section identifies any special considerations or techniques related to using the VCADCS file.

1. The observations for the VCADCS file are built from the data elements of the VCADAA file at the DETAIL timespan.
2. The purpose of the file is to provide an automatic summarized view of the detail data organized around the SMS construct of DATACLAS (CA MICS element STORDATC). Since STORDATC is not maintained as a sequence/summary element in any other VCA file, this file (VCADCS) is the only place in which to view the historical usage of DATACLAS. For example, you can examine this file to view which accounts in your enterprise are using DATACLAS and by careful analysis of the VCADCS file, you may be able to gain new insights into what the ACS routine for DATACLAS is missing. By expanding the boundaries of the DATACLAS ACS routine to include more data allocations or to become less restrictive, the end-user community is freed from coding JCL and the goals of centralized storage administration are furthered.
3. Note that data element DAACOST is carried forward into the VCADCS file through normal CA MICS summarization of numeric elements. If you are charging for DASD at the DAYS level, DAACOST will have missing values. Yet if you are charging at the DETAIL timespan, DAACOST will be summarized by DATACLAS in this file and it provides simply another measure of utilization by DATACLAS.

#### 5.2.4.4 VCADCS Retrieval Examples

In the examples, a SAS macro variable is used to specify the DDname part of the CA MICS file name. These macro variables are a standard part of CA MICS and are available for all files. The macro variable name has the form &diit, where d is the database identifier, ii is the information area name, and t is the timespan. For the examples, a database identifier of P is used. The identifier is installation dependent, so you should find out what the identifiers are at your installation.

1. List the number of data sets and amount of space allocated under different Data Class Names.

```
%LET BY=STORDATC;
%LET BREAK=STORDATC;
PROC SORT DATA=&PVCAD..VCADCS01 OUT=FILE1;
BY &BY;
DATA;
SET FILE1;
%DCSSUM;
RUN;
PROC PRINT;
VAR STORDATC DAADSNNO DAAALLSP;
RUN;
```

## 5.3 DFHSM Inventory Primary Information Area Files

This section identifies each file in the DFSMS Inventory Primary Information Area and defines its levels of summarization and data sequencing. Also, it presents the list of available data elements for each file.

The files maintained in the DFHSM Inventory Primary Information Area include:

- 1 - Migrated Data File (HSM MIG)
- 2 - Volume Planning File (HSM VOL)
- 3 - Tape Planning File (HSM TAP)

### 5.3.1 Migrated Data File (HSMMIG)

The Migrated Data File describes the SMS Class attributes, HSM level, space and current status for each data set in the installation that has been migrated from primary DASD.

This file is derived from data collected by VCC. One observation is built for each migrated data set at HSM Level 1 or Level 2.

The following sections describe the file's organization, list the data elements maintained, and provide usage hints.

- 1 - HSMMIG File Organization
- 2 - HSMMIG Data Elements List
- 3 - HSMMIG Usage Considerations
- 4 - HSMMIG Retrieval Examples

#### 5.3.1.1 HSMMIG File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	HSMACT1	HSMACT2	HSMACT3	DEVCLASS	
	HSMLEVEL	STORSTGC	STORMGTC	YEAR	MONTH	
	DAY	HOURL	DSNAME	ENDTS		
DAYS	SYSID	HSMACT1	HSMACT2	HSMACT3	DEVCLASS	
	HSMLEVEL	STORSTGC	STORMGTC	YEAR	MONTH	
	DAY	HOURL				
WEEKS	N/A					
MONTHS	SYSID	HSMACT1	HSMACT2	HSMACT3	DEVCLASS	
	HSMLEVEL	STORSTGC	STORMGTC	YEAR	MONTH	
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-8. HSMMIG Time-Span Granularity Chart

### 5.3.1.2 HSMMIG Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

**TIMESPAN:** Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL  
D - DAYS  
W - WEEKS  
M - MONTHS  
Y - YEARS  
T - TABLES AREA  
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

**DATA ELEMENT:** The data element name.

**DATA ELEMENT DESCRIPTION:** The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

**GENERATION DATE:** Tue, May 12, 2009

**Note:** Essential data elements are identified by an "E" under the Timespan asterisk (\*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

#### Sequence/Summary Data Elements

XD...E DAY - Day of Month  
XD.M.E DEVCLASS - Device Class  
X....E DSNAME - Data Set Name  
XD...E HOUR - Hour of Day  
XD.M.E HSMACT1 - DIVISION  
XD.M.E HSMACT2 - PROJECT  
XD.M.E HSMACT3 - USER

XD.M..E HSMLEVEL - DFHSM Migration Level (0, 1 or 2)  
XD.M..E MONTH - Month of Year  
XD.M..E STORMGTC - DFSMS Management Class Name  
XD.M..E STORSTGC - DFSMS Storage Class Name  
XD.M..E SYSID - System Identifier  
XD....E WEEK - Week of Year  
XD.M..E YEAR - Year of Century

Common Data Elements

X....E BLKSIZE - Data Set Physical Block Size  
X....E CREATEDT - Creation Date  
XD....E DAYNAME - Name of Day of Week  
X....E DSORG - Data Set Organization  
XD.M..E DURATION - Recording Interval Time  
XD.M..E ENDTS - End Time Stamp  
X....E EXPDT - EXPIRATION DATE  
XD.M..E HSMSAMPL - Count of Sample (Inventory) Runs  
X....E LRECL - Data Set Logical Record Length  
XD.M..E MICSVER - CA MICS Version Number  
X....E ORGSYSID - Originating System Identification  
X....E RECFM - Data Set Record Format  
X....E STORDATC - DFSMS Data Class Name

Accumulated Data Elements

XD.M..E MIGALLSP - Migration Copy Size (MBytes)  
XD.M..E MIGCDSIZ - Compressed Data Set Size (Bytes)  
XD.M..E MIGCDSKB - Compressed Data Set Size (KBytes)  
XD.M..E MIGCDSMB - Compressed Data Set Size (MBytes)  
XD.M.. MIGCHANG - Changed Since Last Backup Count  
XD.M..E MIGCMPDS - Compressed Data Set Count  
XD.M..E MIGCOST - Space\*Time Charges (Cost)  
XD.M.. MIGDSIZE - Migration Copy Size (IBM Kilobytes)  
XD.M.. MIGDSNMT - Empty Data Set Flag  
XD.M..E MIGDSNNO - Count of Migrated Data Sets  
XD.M..E MIGDURSP - Amount of Space\*Time (MBhours)  
XD.M..E MIGESTSP - RECALL Estimated Space (KBytes)  
XD.M.. MIGGDS - Generation Data Set Count  
XD.M.. MIGMIGNO - Number of Times Migrated  
XD.M..E MIGORGSP - Original Allocated Space (KBytes)  
XD.M.. MIGPDSE - PDSE Data Set Count  
XD.M.. MIGRACFD - RACF-Indicated Data Set Count  
XD.M.. MIGREBLK - Reblockable Data Set Count  
XD.M.. MIGSMSID - System Managed Data Set Count  
XD.M..E MIGUDSIZ - Uncompressed Data Set Size (Bytes)  
XD.M..E MIGUDSKB - Uncompressed Data Set Size (KBytes)  
XD.M..E MIGUDSMB - Uncompressed Data Set Size (MBytes)  
XD.M..E MIGUSESP - Original Used Space (KBytes)

**Minimum Data Elements**

XD.M.. MIGMBTS - Min Last Backup Date Time-Stamp  
XD.M.. MIGMNTS - Min Migration Time-Stamp  
XD.M.. MIGREFDT - Oldest Last Reference Date

**Maximum Data Elements**

XD.M.. MIGXASP - Max Migration Copy Size (MBytes)  
XD.M..E MIGRATTS - Max Migration Time-Stamp

**Derived Data Elements**

XD.M..E MIGAVASP - Avg Migration Copy Size (MBytes)  
XD.M..E MIGAVDSP - Avg Megabyte\*Hours (MBhours)

### 5.3.1.3 HSMMIG Usage Considerations

This section identifies one special consideration or technique related to using the HSMMIG file.

An HSM exit, HSMACRT, is available to provide for the mapping of any data element in the Migrated Data File, such as data set name, to user-defined installation account codes. With this addition to the HSM data and the inclusion of CA MICS Accounting and Chargeback, HSM accounting becomes possible.

### 5.3.1.4 HSMMIG Retrieval Examples

In the examples, a SAS macro variable is used to specify the DDname part of the CA MICS file name. These macro variables are a standard part of CA MICS and are available for all files. The macro variable name has the form &diit, where d is the database identifier, ii is the information area name, and t is the timespan. For the examples, a database identifier of P is used. The identifier is installation dependent, so you should find out what the identifiers are at your installation.

1. List all data sets for TSO user TPR030 that have been migrated from primary DASD since the beginning of the year.

```
DATA;  
SET &PHSMX..HSMMIG01;  
IF DSNAME='TPR030';  
IF YEAR(DATEPART(MIGRATTS)) = YEAR(TODAY());  
PROC PRINT; VAR DSNAME MIGRATTS MIGDSIZE;
```

2. Show the amount of data migrated by management class. In this example, we have included the JCL (procedure) you should use to run the SAS job. In this example, the small "x" in the EXEC statement and the PROC SORT statement would need to be replaced by the identifier of the unit data base you are accessing (previous examples have used "P" for the value of "x"). Also note the use of the MIGSUM macro providing summarization for the HSMMIG file.

```
//MIG1    JOB ....  
//REPORT EXEC MICSSHRx  
//SYSIN  DD *  
    %LET BY = STORMGTC;  
    %LET BREAK = STORMGTC;  
    PROC SORT DATA=&xVCAD..HSMMIG01 OUT=SUMFILE;  
        BY &BY;  
    DATA;  
        SET SUMFILE;  
    %MIGSUM;  
    PROC PRINT NOOBS LABEL;  
        VAR STORMGTC MIGDSNNO MIGALLSP;  
    TITLE1 'MIGRATION SUMMARY BY MANAGEMENT CLASS';  
/*
```

An example of the output produced follows. The first detail

line represents 2471 data sets that do not have a value for MGMTCLAS (STORMGTC = ' ').

MIGRATION SUMMARY BY MANAGEMENT CLASS

DFSMS Management Class Name	Count of Migrated Data Sets	Migration Copy Size (MBytes)
	2471	1732.0
DUMP	213	2805.9
LARGE	1213	12635.1
NOMIG	1	0.3
STANDARD	18088	9094.1

Figure 5-9. Sample Migration Summary Report

### 5.3.2 Volume Planning File (HSMVOL)

The Volume Planning File describes the characteristics of DASD volumes before and after daily HSM space management, including percentage occupancy, target and trigger occupancy values, and excess eligible data occupancy.

The following sections describe the file's organization, list the data elements maintained, and provide usage hints.

- 1 - HSMVOL File Organization
- 2 - HSMVOL Data Elements List
- 3 - HSMVOL Usage Considerations
- 4 - HSMVOL Retrieval Examples

### 5.3.2.1 HSMVOL File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	VOLSER	HSMLEVEL	YEAR	MONTH	
	DAY	HOUR	ENDTS			
DAYS	SYSID	VOLSER	HSMLEVEL	YEAR	MONTH	
	DAY	HOUR				
WEEKS	N/A					
MONTHS	SYSID	VOLSER	HSMLEVEL	YEAR	MONTH	
YEARS	SYSID	VOLSER	HSMLEVEL	YEAR		
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-10. HSMVOL Time-Span Granularity Chart

### 5.3.2.2 HSMVOL Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

**TIMESPAN:** Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL  
D - DAYS  
W - WEEKS  
M - MONTHS  
Y - YEARS  
T - TABLES AREA  
. - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

**DATA ELEMENT:** The data element name.

**DATA ELEMENT DESCRIPTION:** The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

**GENERATION DATE:** Tue, May 12, 2009

**Note:** Essential data elements are identified by an "E" under the Timespan asterisk (\*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

#### Sequence/Summary Data Elements

XD...E DAY - Day of Month  
XD...E HOUR - Hour of Day  
XD.MY.E HSMLEVEL - DFHSM Migration Level (0, 1 or 2)  
XD.M..E MONTH - Month of Year  
XD.MY.E SYSID - System Identifier  
XD.MY.E VOLSER - Volume Serial Number  
XD...E WEEK - Week of Year

XD.MY.E YEAR - Year of Century

Common Data Elements

XD....E DAYNAME - Name of Day of Week  
XD.MY.E DURATION - Recording Interval Time  
XD.MY.E ENDTS - End Time Stamp  
XD.MY.E HSMSAMPL - Count of Sample (Inventory) Runs  
XD.MY.E MICSVER - CA MICS Version Number  
X....E ORGSYSID - Originating System Identification

Retained Data Elements

XD.MY.E VOLCAPMB - Total Capacity of Volume in Megabytes  
X....E VOLCOLDT - Date HSM Statistics Were Collected  
XD.MY. VOLTOTAL - Capacity of Volume (IBM Kilobytes)

Accumulated Data Elements

XD.MY. VOLIMGCT - Count of Interval Migration Runs  
XD.MY. VOLIMGOK - Count of Interval Migration Successes  
XD.MY.E VOLPCAFT - Occupancy After Daily Space Management  
XD.MY.E VOLPCBEF - Occupancy Before Daily Space Management  
XD.MY.E VOLPCEED - Excess Eligible Data Occupancy  
XD.MY.E VOLPCHI - High-Threshold (Trigger Occupancy)  
XD.MY.E VOLPCLOW - Low-Threshold (Target Occupancy)  
XD.MY.E VOLVOLNO - Count of Days VOLSER Had Activity

Minimum Data Elements

.D.MY.E VOLMNAFT - Occupancy After Daily Space Management  
.D.MY.E VOLMNBEF - Occupancy Before Daily Space Management  
.D.MY.E VOLMNEED - Excess Eligible Data Occupancy  
.D.MY.E VOLMNHI - High-Threshold (Trigger Occupancy)  
.D.MY.E VOLMNLOW - Low-Threshold (Target Occupancy)

Maximum Data Elements

.D.MY.E VOLMXAFT - Occupancy After Daily Space Management  
.D.MY.E VOLMXBEF - Occupancy Before Daily Space Management  
.D.MY.E VOLMXEED - Excess Eligible Data Occupancy  
.D.MY.E VOLMXHI - High-Threshold (Trigger Occupancy)  
.D.MY.E VOLMXLOW - Low-Threshold (Target Occupancy)

Derived Data Elements

XD.MY.E VOLAFAFT - Occupancy After Daily Space Management  
XD.MY.E VOLAVBEF - Occupancy Before Daily Space Management  
XD.MY.E VOLAVEED - Excess Eligible Data Occupancy

XD.MY.E VOLAVHI - High-Threshold (Trigger Occupancy)  
XD.MY.E VOLAVLOW - Low-Threshold (Target Occupancy)  
XD.MY.E VOLAVXED - Average Total Occupancy Requirement

### 5.3.2.3 HSMVOL Usage Considerations

This section identifies any special considerations or techniques related to using the HSMVOL file.

#### Granularity at DETAIL Timespan

Data for this file comes from VSR records (volume statistics records) maintained in the DFHSM MCDS. MIGRATIONCLEANUPDAYS is a DFHSM SETSYS parameter that controls how many days these VSR records are kept in the MCDS.

When VCC takes an inventory of DFHSM, the ARCUTIL interface provides all the VSR records back to VCC. Thus, if MIGRATIONCLEANUPDAYS is 5, each inventory will result in 5 observations for VOLSER vvvvvv with VOLCOLDT set to a different date for each of the 5 prior days of activity by DFHSM.

The next time that VCC takes an inventory, the 5 most recent days are collected and represented in HSMVOL. The oldest of the 5 days in the prior run has been dropped off. This aging process is similar to GDG limits. The following PROC PRINT shows a typical set of data for this file at the DETAIL timespan.

NOTE: These multiple observations for a volume at DETAIL are summarized into a single observation at DAYS.

DETAIL TIME SPAN VIEW OF HSMVOL FILE											12:14 TUESDAY, mmmddyyyy
----- Volume Serial Number=LRG004 Date HSM Statistics Were Collected=04MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
1	LRG004	ddmmmyy	0	1892.01	82.00 %	10.00 %	80.00 %	82.00 %	82.00 %	0.00 %	1
----- Volume Serial Number=LRG004 Date HSM Statistics Were Collected=06MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
2	LRG004	ddmmmyy	0	1892.01	79.00 %	10.00 %	80.00 %	79.00 %	79.00 %	0.00 %	1
----- Volume Serial Number=LRG004 Date HSM Statistics Were Collected=07MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
3	LRG004	ddmmmyy	0	1892.01	91.00 %	10.00 %	80.00 %	91.00 %	91.00 %	0.00 %	1
----- Volume Serial Number=LRG004 Date HSM Statistics Were Collected=08MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
4	LRG004	ddmmmyy	0	1892.01	91.00 %	10.00 %	80.00 %	91.00 %	91.00 %	0.00 %	1
----- Volume Serial Number=LRG004 Date HSM Statistics Were Collected=09MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
5	LRG004	ddmmmyy	0	1892.01	85.00 %	10.00 %	80.00 %	85.00 %	85.00 %	0.00 %	1
----- Volume Serial Number=LRG004 Date HSM Statistics Were Collected=10MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
6	LRG004	ddmmmyy	0	1892.01	84.00 %	10.00 %	80.00 %	84.00 %	84.00 %	0.00 %	1
----- Volume Serial Number=LRG004 Date HSM Statistics Were Collected=11MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
7	LRG004	ddmmmyy	0	1892.01	84.00 %	10.00 %	80.00 %	84.00 %	84.00 %	0.00 %	1
----- Volume Serial Number=LRG004 Date HSM Statistics Were Collected=13MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
8	LRG004	ddmmmyy	0	1892.01	83.00 %	10.00 %	80.00 %	83.00 %	83.00 %	0.00 %	1
					-----						
					679.00 %	80.00 %	640.00 %	679.00 %	679.00 %	0.00 %	8
-----											
DETAIL TIME SPAN VIEW OF HSMVOL FILE											12:14 TUESDAY, mmmddyyyy
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=04MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
9	LRG005	ddmmmyy	0	1892.01	79.00 %	10.00 %	80.00 %	79.00 %	79.00 %	0.00 %	1
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=05MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
10	LRG005	ddmmmyy	0	1892.01	77.00 %	10.00 %	80.00 %	77.00 %	81.00 %	0.00 %	1
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=06MAYyy -----											
OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
11	LRG005	ddmmmyy	0	1892.01	81.00 %	10.00 %	80.00 %	81.00 %	81.00 %	0.00 %	1
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=07MAYyy -----											

5.3 DFHSM Inventory Primary Information Area Files

OBS	VOLSER	VOLCOLDT	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO
12	LRG005	ddmmyy	0	1892.01	83.00 %	10.00 %	80.00 %	83.00 %	83.00 %	0.00 %	1
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=08MAYyy -----											
13	LRG005	ddmmyy	0	1892.01	85.00 %	10.00 %	80.00 %	85.00 %	85.00 %	0.00 %	1
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=09MAYyy -----											
14	LRG005	ddmmyy	0	1892.01	85.00 %	10.00 %	80.00 %	85.00 %	86.00 %	0.00 %	1
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=10MAYyy -----											
15	LRG005	ddmmyy	0	1892.01	82.00 %	10.00 %	80.00 %	82.00 %	83.00 %	0.00 %	1
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=11MAYyy -----											
16	LRG005	ddmmyy	0	1892.01	80.00 %	10.00 %	80.00 %	80.00 %	82.00 %	0.00 %	1
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=13MAYyy -----											
DETAIL TIME SPAN VIEW OF HSMVOL FILE 12:14 TUESDAY, mmmddyyyy 2											
----- Volume Serial Number=LRG005 Date HSM Statistics Were Collected=13MAYyy -----											
17	LRG005	ddmmyy	0	1892.01	85.00 %	10.00 %	80.00 %	85.00 %	85.00 %	0.00 %	1
					-----	-----	-----	-----	-----	-----	-----
VOLSER					737.00 %	90.00 %	720.00 %	737.00 %	745.00 %	0.00 %	9
					=====	=====	=====	=====	=====	=====	=====
					1416.00 %	170.00 %	1360.00 %	1416.00 %	1424.00 %	0.00 %	17

Figure 5-11. Sample Detail Timespan View of HSMVOL File

DAYS TIME SPAN VIEW OF HSMVOL FILE											12:14 TUESDAY, mmmddyyyy	4
----- Volume Serial Number=LRG004 -----												
OBS	VOLSER	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO		
1	LRG004	0	1892.01	84.87 %	10.00 %	80.00 %	84.87 %	84.87 %	0.00 %	8		
----- Volume Serial Number=LRG005 -----												
OBS	VOLSER	HSMLEVEL	VOLCAPMB	VOLAVXED	VOLAVLOW	VOLAVHI	VOLAVAFT	VOLAVBEF	VOLAVEED	VOLVOLNO		
2	LRG005	0	1892.01	81.88 %	10.00 %	80.00 %	81.88 %	82.77 %	0.00 %	9		
					=====	=====	=====	=====	=====	=====	=====	
					166.76 %	20.00 %	160.00 %	166.76 %	167.65 %	0.00 %	17	

Figure 5-12. Sample Days Timespan View of HSMVOL File

### 5.3.2.4 HSMVOL Retrieval Examples

In the examples, a SAS macro variable is used to specify the DDname part of the CA MICS file name. These macro variables are a standard part of CA MICS and are available for all files. The macro variable name has the form &diit, where d is the database identifier, ii is the information area name, and t is the timespan. For the examples, a database identifier of P is used. The identifier is installation dependent, so you should find out what the identifiers are at your installation.

1. Show all observations for VOLSERS LRG004 and LGR005 in the DETAIL time-span to illustrate the special granularity shown in the figure above.

```
DATA WORK.VOLUMES;
SET &PHSMX..HSMVOL01;
IF VOLSER = 'LRG004' OR VOLSER = 'LRG005';
RUN;
PROC SORT DATA=WORK.VOLUMES; BY VOLSER VOLCOLDT;
PROC PRINT DATA=WORK.VOLUMES;
TITLE 'DETAIL TIME SPAN VIEW OF HSMVOL FILE';
BY VOLSER VOLCOLDT;
SUM VOLVOLNO
VOLAVXED VOLAVLOW VOLAVHI VOLAVAFT
VOLAVBEF VOLAVEED;
VAR VOLSER VOLCOLDT
HSMLEVEL
VOLCAPMB
VOLAVXED VOLAVLOW VOLAVHI VOLAVAFT
VOLAVBEF VOLAVEED;
SUMBY VOLSER;
RUN;
DATA WORK.VOLUME;
SET &PHSMD..HSMVOL01;
IF VOLSER = 'LRG004' OR VOLSER = 'LRG005';
RUN;
PROC SORT DATA=WORK.VOLUME; BY VOLSER;
PROC PRINT DATA=WORK.VOLUME;
TITLE 'DAYS TIME SPAN VIEW OF HSMVOL FILE';
BY VOLSER;
SUM VOLVOLNO
VOLAVXED VOLAVLOW VOLAVHI VOLAVAFT
VOLAVBEF VOLAVEED;
VAR VOLSER
HSMLEVEL
VOLCAPMB
```

```
VOLAVXED VOLAVLOW VOLAVHI VOLAVAFT
VOLAVBEF VOLAVEED;
SUMBY VOLSER;
RUN;
```

2. Display the average daily occupancy rate before and after daily space management during the current month for all volumes beginning with 'TSM'.

```
DATA;
SET &PHSMM..HSMVOL00;
IF VOLSER='TSM';
PROC PRINT; VAR VOLSER VOLAVBEF VOLAVAFT;
```

3. Storage group is not provided by data collector for the HSMVOL file. Combine the data from the VCAVOA file (DETAIL time-span) with the HSMVOL file (DAYS time-span) for yesterday and print various key totals and averages by STORNAME (storage group). Both files are in the P unit data base.

```
PROC SORT DATA=&PVCAX..VCAVOA01
          OUT=WORK.PRIMARY;
BY VOLSER;
PROC SORT DATA=&PHSMD..HSMVOL01
          OUT=WORK.HSM;
BY VOLSER;
DATA WORK.VOLUMES;
MERGE WORK.PRIMARY
      WORK.HSM;
BY VOLSER;
VOAALLMB = (VOASPACA*VOATKCAP)/1000000;
RUN;
PROC SORT DATA=WORK.VOLUMES; BY STORNAME VOLSER;
PROC PRINT LABEL DATA=WORK.VOLUMES;
BY STORNAME;
SUM VOACAPMB VOAALLMB VOLCAPMB;
VAR VOLSER
    HSMLEVEL
    VOACAPMB VOAALLMB
    VOLAVXED VOLAVLOW VOLAVHI VOLAVAFT
    VOLAVBEF VOLAVEED;
SUMBY STORNAME;
RUN;
```

### 5.3.3 Tape Planning File (HSMTAP)

The Tape Planning File quantifies the tapes under control of DFHSM according to 3 types or categories:

- o tapes used for full volume dumps
- o tapes used for incremental backups
- o tapes used for migration level 2

For each of the above 3 categories of tapes managed by DFHSM, the following counts are provided:

- o the number of full tapes
- o the number of partially full tapes
- o the number of empty tapes

The following sections describe the file's organization, list the data elements maintained, and provide usage hints.

- 1 - HSMTAP File Organization
- 2 - HSMTAP Data Elements List
- 3 - HSMTAP Usage Considerations
- 4 - HSMTAP Retrieval Examples

### 5.3.3.1 HSMTAP File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	YEAR	MONTH	DAY	HOUR	
	ENDTS					
DAYS	SYSID	YEAR	MONTH	DAY	HOUR	
WEEKS	SYSID	YEAR	WEEK			
MONTHS	SYSID	YEAR	MONTH			
YEARS	SYSID	YEAR				
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-13. HSMTAP Time-Span Granularity Chart

### 5.3.3.2 HSMTAP Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

**TIMESPAN:** Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

**DATA ELEMENT:** The data element name.

**DATA ELEMENT DESCRIPTION:** The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

**GENERATION DATE:** Tue, May 12, 2009

**Note:** Essential data elements are identified by an "E" under the Timespan asterisk (\*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

#### Sequence/Summary Data Elements

XD...E	DAY	- Day of Month
XD...E	HOURL	- Hour of Day
XD.M..E	MONTH	- Month of Year
XDWMY.E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWMY.E	YEAR	- Year of Century

Common Data Elements

XD...E DAYNAME - Name of Day of Week  
XDWMY.E DURATION - Recording Interval Time  
XDWMY.E ENDTS - End Time Stamp  
XDWMY.E HSMSAMPL - Count of Sample (Inventory) Runs  
XDWMY.E MICSVER - CA MICS Version Number  
X....E ORGSYSID - Originating System Identification

Accumulated Data Elements

XDWMY.E TAPBFULL - Number of Backup Tapes (Full)  
XDWMY.E TAPBNULL - Number of Backup Tapes (Empty)  
XDWMY.E TAPBPART - Number of Backup Tapes (Partial)  
XDWMY.E TAPDFULL - Number of Dump Tapes (Full)  
XDWMY.E TAPDNULL - Number of Dump Tapes (Empty)  
XDWMY.E TAPDPART - Number of Dump Tapes (Partial)  
XDWMY.E TAPMFULL - Number of Migration Tapes (Full)  
XDWMY.E TAPMNULL - Number of Migration Tapes (Empty)  
XDWMY.E TAPMPART - Number of Migration Tapes (Partial)

### 5.3.3.3 HSMTAP Usage Considerations

This section identifies any special considerations or techniques related to using the HSMTAP file.

Suggest You Keep Many Cycles

Generating support for many cycles of the HSMTAP file will enable you to graph the change in tape utilization by DFHSM over a long period of time.

This file contains only one small observation per inventory of DFHSM. Normally, this inventory will only occur once per day. Therefore, because of this extremely low resource consumption we recommend you keep many cycles of this file at either the DETAIL or DAYS time-span. If you choose DAYS for the location of the many cycles, then DETAIL should be set to only a few cycles (and vice versa).

### 5.3.3.4 HSMTAP Retrieval Examples

In the examples, a SAS macro variable is used to specify the DDname part of the CA MICS file name. These macro variables are a standard part of CA MICS and are available for all files. The macro variable name has the form &diit, where d is the database identifier, iii is the information area name, and t is the timespan. For the examples, a database identifier of P is used. The identifier is installation dependent, so you should find out what the identifiers are at your installation.

1. Display the number of tapes under DFHSM that were either full or partially full of HSM Level 2 migration data as of yesterday.

```
PROC PRINT DATA=&PHSMX..HSMTAP01;
VAR SYSID YEAR MONTH TAPMFULL TAPMPART;
```

2. Plot the change in usage of DFHSM-owned backup tapes over the past 4 days. An output sample follows the code.

```
DATA;
  SET &PHSMD..HSMTAP01 &PHSMD..HSMTAP02
    &PHSMD..HSMTAP03 &PHSMD..HSMTAP04;
  DATE = DATEPART(ENDTS);
  FORMAT DATE DATE7.;
PROC PLOT NOLEGEND;
  PLOT TAPBFULL*DATE = 'F' TAPBNULL*DATE = 'E'
    TAPBPART*DATE = 'P' / OVERLAY;
  LABEL TAPBFULL = 'NUMBER OF TAPES';
  TITLE1 'HSM BACKUP TAPE USAGE';
  FOOTNOTE1 'F = FULL BACKUP VOLUMES  ';
  FOOTNOTE2 'P = PARTIAL BACKUP VOLUMES';
  FOOTNOTE3 'E = EMPTY BACKUP VOLUMES  ';
```

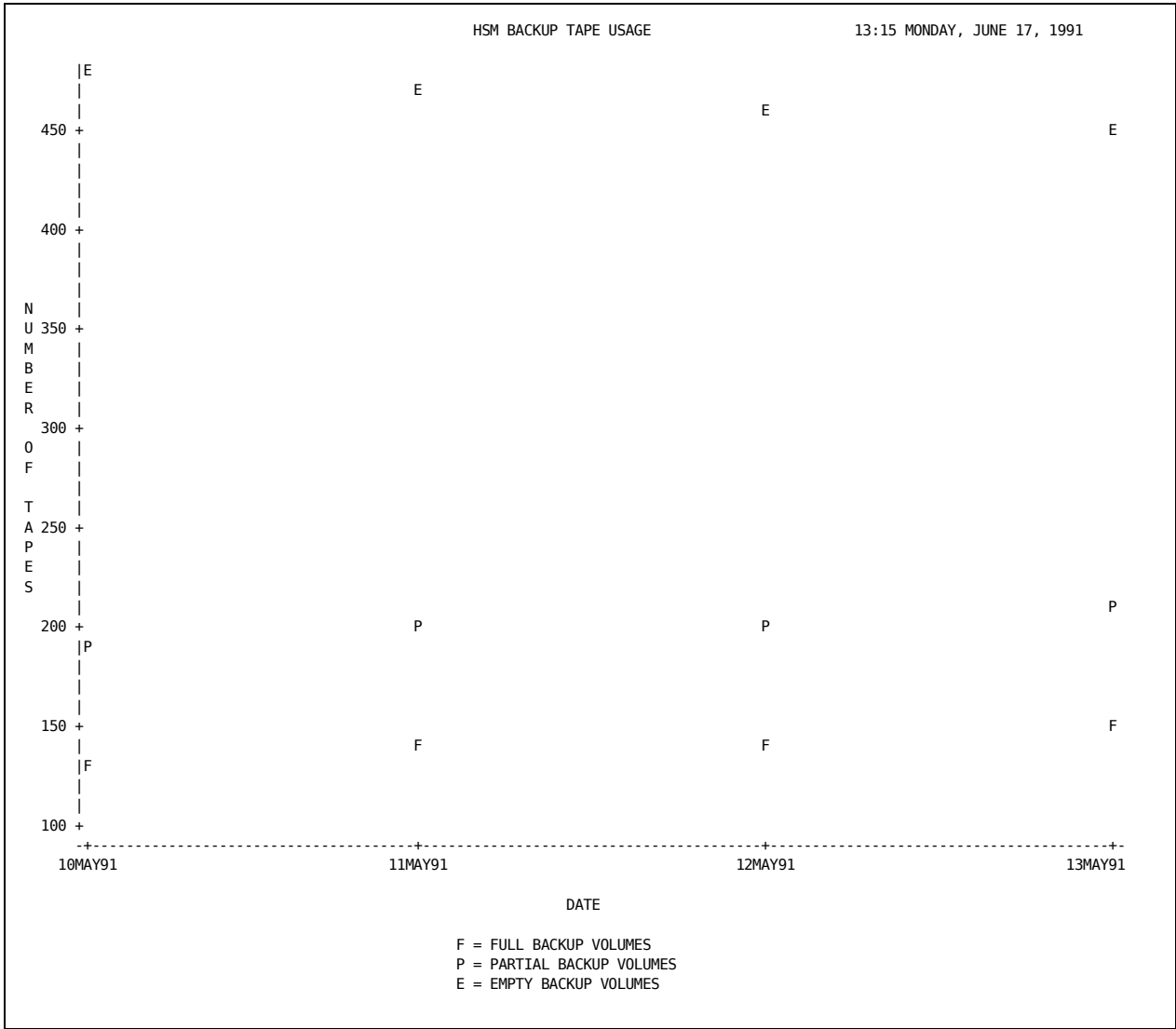


Figure 5-14. Sample HSM Backup Tape Usage Report

## 5.4 DFHSM Inventory Backup Information Area Files

This section identifies each file in the DFSMS Inventory Backup Information Area and defines its levels of summarization and data sequencing. Also, it presents the list of available data elements for each file.

The files maintained in the DFHSM Inventory Backup Information Area include:

- 1 - BACKUP Data File (HSBBAC)

### 5.4.1 BACKUP Data File (HSBBAC)

At the DETAIL timespan, this file describes a variety of information about the DFHSM-managed backup versions for a given data set at a given point in time.

Records created by VCC are at the backup version level. That is, there is one raw record for each backup version in the inventory maintained by DFHSM. The DAY095 processing reduces multiple backup versions for a data set into one observation for that data set. The fact that this summarization is done in the DETAIL timespan is a slight departure from most CA MICS DAILY processing. It keeps the size of the DETAIL timespan manageable while still offering full reporting capability.

The summarization that takes place from DETAIL to DAYS then eliminates DSNNAME as a sequence variable and reduces the data regarding backup resources to the account structure or cost center variables.

The following sections describe the file's organization, list the data elements maintained, and provide usage hints.

- 1 - HSBBAC File Organization
- 2 - HSBBAC Data Elements List
- 3 - HSBBAC Usage Considerations
- 4 - HSBBAC Retrieval Examples

#### 5.4.1.1 HSBBAC File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	HSMACT1	HSMACT2	HSMACT3	DEVCLASS	
	STORSTGC	STORMGTC	YEAR	MONTH	DAY	
	HOUR	DSNAME	ENDTS			
DAYS	SYSID	HSMACT1	HSMACT2	HSMACT3	DEVCLASS	
	STORSTGC	STORMGTC	YEAR	MONTH	DAY	
	HOUR					
WEEKS	N/A					
MONTHS	SYSID	HSMACT1	HSMACT2	HSMACT3	DEVCLASS	
	STORSTGC	STORMGTC	YEAR	MONTH		
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-15. HSBAC Time-Span Granularity Chart

### 5.4.1.2 HSBAC Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

**TIMESPAN:** Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

X - DETAIL  
 D - DAYS  
 W - WEEKS  
 M - MONTHS  
 Y - YEARS  
 T - TABLES AREA  
 . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

**DATA ELEMENT:** The data element name.

**DATA ELEMENT DESCRIPTION:** The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

**GENERATION DATE:** Tue, May 12, 2009

**Note:** Essential data elements are identified by an "E" under the Timespan asterisk (\*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

#### Sequence/Summary Data Elements

XD...E DAY - Day of Month  
 XD.M..E DEVCLASS - Device Class  
 X....E DSNAME - Data Set Name  
 XD...E HOUR - Hour of Day  
 XD.M..E HSMACT1 - DIVISION  
 XD.M..E HSMACT2 - PROJECT

XD.M..E HSMACT3 - USER  
XD.M..E MONTH - Month of Year  
XD.M..E STORMGTC - DFSMS Management Class Name  
XD.M..E STORSTGC - DFSMS Storage Class Name  
XD.M..E SYSID - System Identifier  
XD....E WEEK - Week of Year  
XD.M..E YEAR - Year of Century

Common Data Elements

X....E BLKSIZE - Data Set Physical Block Size  
XD....E DAYNAME - Name of Day of Week  
X....E DSORG - Data Set Organization  
XD.M..E DURATION - Recording Interval Time  
XD.M..E ENDTS - End Time Stamp  
XD.M..E HSMSAMPL - Count of Sample (Inventory) Runs  
XD.M..E MICSVER - CA MICS Version Number  
X....E ORGSYSID - Originating System Identification  
X....E RECFM - Data Set Record Format  
X....E STORDATC - DFSMS Data Class Name

Accumulated Data Elements

XD.M..E BACALLSP - Total Size of Backup Versions (MBytes)  
XD.M.. BACCATLG - Count of Cataloged When Backup Made  
XD.M..E BACCDSIZ - Compressed Data Set Size (Bytes)  
XD.M..E BACCDSKB - Compressed Data Set Size (KBytes)  
XD.M..E BACCDSMB - Compressed Data Set Size (MBytes)  
XD.M..E BACCPDS - Compressed Data Set Count  
XD.M..E BACCOST - Space\*Time Charges (Cost)  
XD.M..E BACDSNNO - Number of Data Sets Backed Up  
XD.M..E BACDURSP - Amount of Space\*Time (MBhours)  
XD.M..E BACESTSP - RECOVER Estimated Space (KBytes)  
XD.M.. BACGDS - Generation Data Set Count  
XD.M.. BACNOENQ - Count of Backups Taken Without ENQ  
XD.M.. BACOPEN - Count of Backup-While-OPEN Candidates  
XD.M..E BACORGSP - Original Allocated Space (KBytes)  
XD.M.. BACPDSE - PDSE Data Set Count  
XD.M.. BACRACFD - RACF-Indicated Data Set Count  
XD.M.. BACREBLK - Reblockable Data Set Count  
XD.M.. BACSMSID - System Managed Data Set Count  
XD.M..E BACUDSIZ - Uncompressed Data Set Size (Bytes)  
XD.M..E BACUDSKB - Uncompressed Data Set Size (KBytes)  
XD.M..E BACUDSMB - Uncompressed Data Set Size (MBytes)  
XD.M..E BACUSESP - Original Used Space (KBytes)  
XD.M..E BACVERNO - Number of Backup Versions for DSNAME

Minimum Data Elements

XD.M..E BACMNASP - Minimum Backup Version Size (MBytes)  
XD.M..E BACMNTS - Timestamp of Oldest Backup

Maximum Data Elements

XD.M..E BACMXASP - Maximum Backup Version Size (MBytes)  
XD.M..E BACMXTS - Timestamp of Newest Backup

Derived Data Elements

XD.M..E BACAVASP - Avg Backup Size (MBytes)  
XD.M..E BACAVDSP - Avg Megabyte\*Hours (MBhours)

### 5.4.1.3 HSBAC Usage Considerations

This section identifies one special consideration or technique related to using the HSBAC file.

Note that data element BACALLSP at the DETAIL timespan contains the amount of space occupied by all backup versions of a data set. Unlike most other CA MICS files at the DETAIL timespan, the HSBAC file has already gone through a level of summarization. Rather than an observation for each data set backup version, there is one observation per data set, containing information relevant to all data set backup versions.

#### 5.4.1.4 HSBAC Retrieval Examples

In the examples, a SAS macro variable is used to specify the DDname part of the CA MICS file name. These macro variables are a standard part of CA MICS and are available for all files. The macro variable name has the form &diit, where d is the database identifier, ii is the information area name, and t is the timespan. For the examples, a database identifier of P is used. The identifier is installation dependent, so you should find out what the identifiers are at your installation.

1. List backup statistics for SMS management class APFLIB reflecting the utilization of incremental backup facilities last month.

```
DATA;  
SET &PHSBM..HSBBAC01;  
IF STORMGTC='APFLIB';  
PROC PRINT;  
VAR BACMNASP BACMXASP BACAVASP BACCOST;
```

2. Show department level statistics regarding the number of backup versions being maintained by DFHSM.

```
%LET BY = HSMACT3;  
%LET BREAK = HSMACT3;  
PROC SORT DATA=&PHSBD..HSBBAC01 OUT=SUMFILE;  
BY &BY;  
DATA;  
SET SUMFILE;  
%BACSUM;  
PROC PRINT NOOBS LABEL UNIFORM;  
VAR HSMACT3 BACDSNNO BACALLSP BACMNASP  
BACAVASP BACMXASP;  
SUM BACDSNNO BACALLSP;  
LABEL HSMACT3 = 'DEPARTMENT';  
TITLE1 'HSM BACKUP SUMMARY BY DEPARTMENT';
```

HSM BACKUP SUMMARY BY DEPARTMENT						16:07 MONDAY, JUNE 17, 1991	1
DEPARTMENT	Number of Data Sets Backed Up	Total Size of Backup Versions (MBytes)	Minimum Backup Version Size (MBytes)	Avg Backup Size (MBytes)	Maximum Backup Version Size (MBytes)		
AA1	110	3.5	0.01638	0.032	0.573		
ACC	316	857.7	0.01638	2.714	154.465		
ADM	7	0.1	0.01638	0.016	0.016		
AD1	1	0.0	0.01638	0.016	0.016		
AIC	27	7.3	0.01638	0.272	3.129		
ALL	2	0.0	0.01638	0.016	0.016		
ASX	59	630.7	0.01638	10.689	95.926		
ATM	887	1665.0	0.00000	1.877	124.715		
AXC	2	0.1	0.03277	0.033	0.033		
BAC	213	41.2	0.01638	0.194	2.179		
BST	1983	3422.9	0.01638	1.726	116.113		
BUN	1707	1621.9	0.00000	0.950	88.766		
CAP	470	2963.9	0.01638	6.306	414.438		
CAT	2	5.4	0.16383	2.679	5.194		
CA1	888	1076.9	0.01638	1.213	39.121		
CA2	60	1.2	0.01638	0.020	0.066		
CDL	342	781.7	0.01638	2.286	28.406		
CD1	5	0.1	0.01638	0.016	0.016		
CJ1	120	4.1	0.01638	0.034	0.868		
CK1	21	0.7	0.01638	0.032	0.066		
CMP	128	278.4	0.01638	2.175	56.262		
CM1	37	9.5	0.01638	0.255	2.572		
CNM	3	4.2	1.39258	1.393	1.393		
CSS	286	12807.1	0.00000	44.777	948.000		
CST	287	566.4	0.01638	1.973	25.133		
CSU	742	3975.1	0.00000	5.357	198.063		
CS1	188	46.8	0.01638	0.249	14.074		
CT1	108	8.5	0.01638	0.079	1.311		
DA1	2093	796.3	0.01638	0.380	38.535		
DA2	241	30.6	0.01638	0.127	0.934		
DD1	20	1.1	0.01638	0.053	0.704		
DEM	3	0.0	0.01638	0.016	0.016		
DOC	8	0.5	0.01638	0.066	0.131		
DPM	7	40.1	0.01638	5.729	13.992		
DPO	309	238.2	0.01638	0.771	8.962		
DPW	68	8.7	0.01638	0.127	0.639		
DSI	10047	10267.8	0.01638	1.022	170.605		
DSM	1	0.2	0.24576	0.246	0.246		
DST	536	2163.9	0.01638	4.037	236.238		
DSX	4	0.8	0.01638	0.197	0.360		
DXB	10239	16368.4	0.01638	1.598	214.270		
EDU	106	16.2	0.01638	0.153	2.277		
EMG	29	12.0	0.01638	0.414	3.850		
END	34	18.6	0.01638	0.546	1.720		
ESA	2	0.1	0.01638	0.033	0.049		
FDL	502	327.4	0.01638	0.652	26.395		
HSA	1	0.0	0.01638	0.016	0.016		

HSM BACKUP SUMMARY BY DEPARTMENT						16:07 MONDAY, JUNE 17, 1991	2
DEPARTMENT	Number of Data Sets Backed Up	Total Size of Backup Versions (MBytes)	Minimum Backup Version Size (MBytes)	Avg Backup Size (MBytes)	Maximum Backup Version Size (MBytes)		
IAM	311	3122.7	0.01638	10.041	432.750		
IDP	7	1.1	0.06552	0.159	0.344		
IIN	190	234.2	0.01638	1.232	26.902		
INF	9	977.2	0.96664	108.578	267.750		
IPO	274	202.1	0.01638	0.738	27.375		
ISP	5	0.6	0.01638	0.118	0.360		
ISR	12	12.5	0.0164	1.0376	6.717		
ITS	209	304.1	0.0164	1.4551	36.402		
JAP	4	0.1	0.0164	0.0246	0.033		
JOB	241	366.4	0.0164	1.5200	8.716		
LCM	114	153.5	0.0164	1.3464	32.129		
LGT	2	0.0	0.0164	0.0164	0.016		
MA1	146	24.2	0.0164	0.1657	2.539		
MB1	21	0.4	0.0164	0.0211	0.033		
MDG	7	4.9	0.0328	0.7068	2.179		
MD1	7	0.1	0.0164	0.0164	0.016		
MEI	28	5.8	0.0164	0.2054	0.754		
MIC	86	25.3	0.0164	0.2947	1.688		
MIM	259	921.5	0.0164	3.5579	82.605		
MLG	3	0.1	0.0164	0.0273	0.033		
MSP	5	5.2	0.0164	1.0320	2.081		
MSS	498	1370.4	0.0164	2.7517	49.395		
MS6	338	774.2	0.0164	2.2903	77.559		
MVS	36	120.3	0.0164	3.3413	10.715		
NA1	81	6.2	0.0164	0.0764	1.507		
NCP	3	27.0	6.9795	8.9946	10.453		
NET	680	7130.7	0.0164	10.4861	291.563		
NJL	3	0.0	0.0164	0.0164	0.016		
NPG	4	1.7	0.3932	0.4260	0.459		
NPM	1	0.0	0.0492	0.0492	0.049		
NTC	161	180.3	0.0164	1.1199	11.501		
OA1	91	28.0	0.0164	0.3082	14.369		
ONL	213	656.5	0.0164	3.0818	80.199		
OPR	234	855.9	0.0164	3.6577	39.977		
OVN	179	99.8	0.0164	0.5576	10.731		
O1A	1	0.0	0.0164	0.0164	0.016		
PAS	116	1083.7	0.0164	9.3420	146.520		
PER	422	1202.1	0.0000	2.8484	129.660		
POR	9	3.2	0.0164	0.3549	1.442		
PRD	1152	20673.3	0.0000	17.9453	649.750		
QCA	2083	6442.2	0.0164	3.0925	124.777		
QCM	50	16.5	0.0164	0.3296	2.720		
QMF	65	2.2	0.0164	0.0335	0.180		
RCS	67	60.9	0.0164	0.9094	7.029		
SAR	140	193.7	0.0164	1.3831	120.520		
SAS	1	26.6	26.6211	26.6211	26.621		
SA1	179	78.3	0.0164	0.4373	9.568		
SER	4	0.1	0.0164	0.0164	0.016		

HSM BACKUP SUMMARY BY DEPARTMENT						16:07 MONDAY, JUNE 17, 1991	3
DEPARTMENT	Number of Data Sets Backed Up	Total Size of Backup Versions (MBytes)	Minimum Backup Version Size (MBytes)	Avg Backup Size (MBytes)	Maximum Backup Version Size (MBytes)		
SMD	83	33.7	0.0164	0.4064	3.064		
SMR	25	14.9	0.0164	0.5950	4.882		
SMS	26	117.1	0.0164	4.5049	13.861		
STX	292	733.9	0.0164	2.5132	91.551		
SYS	1371	2731.9	0.0164	1.9924	55.195		
TA1	39	1.9	0.0164	0.0496	0.147		
TD1	17	8.7	0.0164	0.5127	2.572		
TEC	152	349.4	0.0164	2.2988	38.008		
TIM	14	0.9	0.0164	0.0655	0.197		
NSM	209	17.1	0.0164	0.0818	0.819		
TPX	344	2406.8	0.0164	6.9963	277.063		
TSO	373	1034	0.000000	2.77295	64.5820		
UIS	7	1	0.016383	0.16383	0.5079		
USE	8	4	0.016383	0.54268	3.2766		
VMD	1	0	0.229370	0.22937	0.2294		
VOS	16	9	0.016383	0.56111	1.5728		
XDC	3	3	0.589813	0.99390	1.6055		
XVS	2	0	0.016383	0.02458	0.0328		
	=====	=====					
	44984	115936					

Figure 5-16. Sample HSM Backup Summary by Department Report

## 5.5 USS File System Information Area File

This section identifies each file in the USS File System Information Area and defines its levels of summarization and data sequencing. Also, it presents the list of available data elements for each file.

The file contained in the File System Information Area is:

- 1 - USS Directory Entry File (HFSDIR)

### 5.5.1 USS Directory Entry File (HFSDIR)

FILENAME=HFSDIR

The USS Directory Entry File describes the attributes, allocation, and current status for each directory in the file system file.

This file is derived from data collected by the USS File Systems Collector. One observation is built for each directory entry in a file system selected for processing.

Sources of data for the observations in this file are as follows:

- o The format-1 DSCB in the VTOC provides the data set name and most of the attributes of the data set.
- o The SMS cells in the WDS provide the SMS class name values for DATACLAS, STORCLAS, and MGMTCLAS constructs if the data set is SMS-managed.
- o The storage group name is obtained by VCC from the SMS address space if available.

The HFSDIR file contains observations for directory entries of file system. Each directory entry carries associated file system name (DSNAME). The file system (subtype 7) records that contain no value for VOLSER are not kept in the CA MICS database. This happens when a file system is scanned that only lives in virtual storage. Since this data never resided on the DASD, its VOLSER value is not assigned.

The DSNAME and DEVTYPE elements in HFSDIR file come from subtype 7 records of raw input. The following sections describe the file's organization, list the data elements maintained, and provide usage hints.

- 1 - HFSDIR File Organization
- 2 - HFSDIR Data Elements List
- 3 - HFSDIR Usage Considerations
- 4 - HFSDIR Retrieval Examples

### 5.5.1.1 HFSDIR File Organization

The table below identifies data elements by which the file is sequenced and summarized in each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	VCAACT1	VCAACT2	VCAACT3	DIRFSTNM	
	DEVTYPE	STORSTGC	STORMGTC	YEAR	MONTH	
	DAY	hour	DSNAME	ENDTS		
DAYS	SYSID	VCAACT1	VCAACT2	VCAACT3	DIRFSTNM	
	DEVTYPE	STORSTGC	STORMGTC	YEAR	MONTH	
	DAY	hour				
WEEKS	N/A					
MONTHS	SYSID	VCAACT1	VCAACT2	VCAACT3	DIRFSTNM	
	DEVTYPE	STORSTGC	STORMGTC	YEAR	MONTH	
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-5. HFSDIR Time-Span Granularity Chart

### 5.5.1.2 HFSDIR Data Elements List

The table below identifies data elements contained in this file. The entries for each data element are:

**TIMESPAN:** Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

**DATA ELEMENT:** The data element name.

**DATA ELEMENT DESCRIPTION:** The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

**GENERATION DATE:** Tue, May 12, 2009

**Note:** Essential data elements are identified by an "E" under the Timespan asterisk (\*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----------------	-----------------	-------------------------------------

#### Sequence/Summary Data Elements

- XD...E DAY - Day of Month
- XD.M..E DEVTYPE - Device Type
- XD.M..E DIRFSTNM - File System Type Name
- X....E DSNAME - Data Set Name
- XD...E HOUR - Hour of Day
- XD.M..E MONTH - Month of Year
- XD.M..E STORMGTC - DFSMS Management Class Name

XD.M..E STORSTGC - DFSMS Storage Class Name  
XD.M..E SYSID - System Identifier  
XD.M..E VCAACT1 - DIVISION  
XD.M..E VCAACT2 - PROJECT  
XD.M..E VCAACT3 - USER  
XD....E WEEK - Week of Year  
XD.M..E YEAR - Year of Century

#### Common Data Elements

X....E BLKSIZE - Data Set Physical Block Size  
XD....E DAYNAME - Name of Day of Week  
XD.M..E DURATION - Recording Interval Time  
XD.M..E ENDTS - End Time Stamp  
XD.M..E MICSVER - CA MICS Version Number  
X....E ORGSYSID - Originating System Identification  
X....E STORDATC - DFSMS Data Class Name  
X....E VOLSER - Volume Serial Number

#### Retained Data Elements

X....E DIRCRETS - Create Date Time Stamp  
XD.M..E DIRFMT - DIR Entry (File) Format  
X....E DIRFNAME - File Name  
X....E DIRGID - Group ID  
X....E DIRLACTS - Last Access Date Time Stamp  
X....E DIRLMOTS - Last Modified Date Time Stamp  
X....E DIRNAME1 - Pathname 1  
X....E DIRNAME2 - Pathname 2  
X....E DIRNAME3 - Pathname 3  
X....E DIRNAME4 - Pathname 4  
X....E DIRNAME5 - Pathname 5  
X....E DIRNAME6 - Pathname 6  
X....E DIROID - Owner ID  
XD.M..E DIRTKCAP - Published Track Capacity in Bytes  
XD.M..E DIRTKCBY - Actual Track Capacity in Bytes  
X....E DIRUID - User ID  
X....E DIRXSCID - External Security Audit-ID

#### Accumulated Data Elements

XD.M..E DIRALLSP - Amount of Space Allocated (MBytes)  
XD.M..E DIRAPFAU - File is APF Authorized Flag  
XD.M..E DIRAUTOM - File System is Automount Flag  
XD.M..E DIRBINRY - File is a Binary File Flag  
XD.M..E DIRBLKSA - Number of Blocks Allocated  
XD.M..E DIRCMPRS - Data is Compressed  
XD.M..E DIRCOST - Space Charges  
XD.M..E DIRDSNNO - Number of Directory Entries

XD.M..E DIRDUALC - Volume Has Dual Copy Active  
XD.M..E DIRDURSP - Amount of Space-Time (MByte-Hours)  
XD.M..E DIREXPRT - File System Was Exported  
XD.M..E DIRKBHR - Amount of Space-Time (KBytes-Hours)  
XD.M..E DIRKBYTA - Amount of Space Allocated (KBytes)  
XD.M..E DIRMNTPT - Directory Name is Mount Point Flag  
XD.M..E DIRMRO - File System Mounted Read-Only  
XD.M..E DIRPGMC - File is Program Controlled Flag  
XD.M..E DIRRAID - Volume Uses RAID Technology  
XD.M..E DIRRONLY - File is Read-Only Flag  
XD.M..E DIRSPACA - Amount of Space Allocated (Tracks)  
XD.M..E DIRSYMBL - File is Symbolic Link Flag  
XD.M..E DIRXSLNK - File Is External Symbolic Link Flag

### 5.5.1.3 HFSDIR Usage Considerations

This section identifies any special considerations or techniques related to using the HFSDIR file.

Relationship to Data Sources from VCC

HFSDIR is built from the data in the USS catalog and contains observations for all files and directories for the mounted file systems that are scanned. The HFS scan is different from the VCC scan in which all data sets residing on the selected volume are scanned. A volume may contain many file systems but HFS scan scans only those file systems that are mounted at the time the scan is started.

An example of the basic information provided for every file or directory in each observation is the space allocated in blocks (DIRBLKSA). The space allocated in blocks is the space used by the file or directory. At the file system level, the total space allocated is reserved when the file system is created but only a portion of that file system may be used. The actual space is utilized by the individual files or directories that are created in the file system. The allocated but unused space would not be used for accounting purposes as it is not occupied by any file or directory. For this reason and the fact that only mounted systems are represented in HFSDIR file, we strongly discourage the use of this file for accounting. The existing VCADAA file contains a single record for each file system data set (mounted and unmounted) reflecting all space allocated.

Since all files and directories of the file systems are scanned, the information lends itself to detail analysis of the individual file and/or directory as well as the entire file system.

#### 5.5.1.4 HFSDIR Retrieval Examples

In the examples, a SAS macro variable is used to specify the DDname part of the CA MICS file name. These macro variables are a standard part of CA MICS and are available for all files. The macro variable name has the form &diit, where d is the database identifier, iii is the information area name, and t is the timespan. For the examples, a database identifier of P is used. The identifier is installation dependent, so you should find out what the identifiers are at your installation.

1. List all scanned file systems created in the last 30 days with a high level data set name qualifier of 'CMPR'.

```
DATA;
SET &PHFSX..HFSDIR01;
IF DSNAME='CMPR';
IF DATEPART(CREATETS) > DATEPART(ENDTS) - 30;
PROC PRINT; VAR DSNAME VOLSER CREATEDT DAASPACA DAASPACU;
```

2. Show all scanned file systems that reside on VOLSER=OMVS01 and the total megabytes they occupy. Report the largest data sets first.

```
DATA ONE;
  SET &PHFSX..HFSDIR01;
  IF VOLSER='OMVS01';
PROC SORT;
  BY DESCENDING DAAALLSP;
PROC PRINT NOOBS UNIFORM N;
  VAR DSNAME DEVTYPE VOLSER CREATETS
      DAAREFTS DAAALLSP;
  SUM DAAALLSP;
RUN;
```

# Chapter 6: DATA SOURCES

---

The CA MICS Space Collector (VCC) provides the data processed by the CA MICS Space Analyzer.

Refer to Section 5.2 of the CA MICS Space Collector Guide for details regarding the record format and data sources used.

Also, for VCA, refer to the INPUT statements used to read the raw data created by VCC. For HSM, refer to the INPUT statements used to read the raw data created by VCC.

This section contains the following topics:

[6.1 CA MICS and CA SMF Director Interface](#) (see page 187)

## 6.1 CA MICS and CA SMF Director Interface

CA SMF Director is an SMF management product with special features that can be used to optimize CA MICS processing. CA SMF Director can significantly reduce operational overhead by creating content specific files while simultaneously creating an SMF historical archive during the SMF dump process.

CA MICS DAILY and incremental update operational jobs can be modified to take advantage of this product, provided your site is licensed for and has CA SMF Director installed in your environment.

CA SMF Director provides functions that extract SMF data at dump time or from previously archived SMF files. These functions eliminate the need for any external utilities used for preprocessing of the SMF dump tape for data extraction prior to execution of the DAILY job. In addition, the DAYSMF step of the CA MICS DAILY job may no longer be required. For details on how to eliminate the DAYSMF step, see section 5.10, Removing the DAYSMF Step from the DAILY Job of the PIOM guide.

CA SMF Director SPLIT and EXTRACT statement operands provide a way for you to create content specific files to meet your requirements. The main functions of the operands include the ability to:

- o Select system identifiers
- o Select or exclude SMF record types and subtypes
- o Use conditional statements for granularity
- o Define time boundaries for SMF data

To use the SPLIT function of CA SMF Director for CA MICS components, one or more SPLIT statements must be coded. The split is performed at SMF dump time and the files can then be used as input to products in one or more units. The same results can be accomplished by using the EXTRACT function of CA SMF Director, which retrieves data from the previously created history files. For more information on the SPLIT and EXTRACT functions, see the CA SMF Director User Guide at <http://ca.com/support>.

A complete set of SMF record types and subtypes, for each CA MICS component that inputs SMF data, can be found in `sharedprefix.MICS.PARMS(cccSMFD)`, where `ccc` is the three-character product identifier. Each PARMS member lists the available SMF record types and subtypes for that component in a format suitable for use in CA SMF Director SPLIT and EXTRACT statements. These members are provided as examples only; the CA SMF Director JCL should not reference `sharedprefix.MICS.PARMS(cccSMFD)`.

# Chapter 7: PARAMETERS

---

Defining the product parameters requires that you gain the necessary understanding of your installation and its needs and translate that understanding into CA MICS parameters.

In particular, this chapter requests that you:

- o Make several policy decisions
- o Complete various worksheets
- o Translate the worksheet entries into the corresponding CA MICS parameter library entries

These activities represent the major portion of the product installation process.

The CA MICS administrator should use this chapter as a detailed reference when installing or modifying the product, along with Chapters 2 and 3 of the PIOM, which document the mechanics of the installation process and include checklists that describe each installation step.

If you have a question at any time during your review of the material presented here, please contact Technical Support.

This section contains the following topics:

[7.1 Environmental Considerations](#) (see page 190)

[7.2 Complex Level Parameters](#) (see page 190)

[7.3 Unit Level Parameters](#) (see page 207)

## 7.1 Environmental Considerations

Before coding product parameters, you need to know about the MVS environment(s) at your installation. Before you specify parameters for the product:

- o Examine existing user groups, account codes, and job control information used on the system. This information helps you code appropriate values for account codes.
- o Review the default options for the parameters to determine their applicability to your site.
- o Review current CA MICS parameter specifications:
  - Review database unit specifications to determine which unit or units should include data from the CA MICS Space Collector (VCC).
  - Review the SYSID parameter to determine whether it correlates to the SMF data. Your site's SYSID definition is located in the prefix.MICS.PARMS data set associated with each database unit.
  - Examine existing accounting standards for other CA MICS products, if applicable, to ensure that CA MICS Accounting and Chargeback aggregates information consistently.

## 7.2 Complex Level Parameters

Complex level parameters describe the CA MICS Space Analyzer to the CA MICS system. These parameters are discussed in the following subsections:

- 1 - Define Account Code Structure (VCAACCT)
- 2 - Account Code Derivation Exit (VCAACRT)
- 3 - Synchronize HSMACCT with VCAACCT
- 4 - Synchronize HSMACRT with VCAACRT
- 5 - Analyzer Definition Statements (VCAGENIN)

## 7.2.1 Define Account Code Structure (VCAACCT)

The Space Analyzer provides nine levels of account codes for you to associate users and user groups with data set ownership. The account codes are used as file (summarization) keys in the following CA MICS files:

- VCADAA - Data Set Allocation File
- VCA\_VS - VSAM Data Set Allocation File
- VCADCS - DATACLAS Summary File
- HSMMIG - Migrated Data Set File
- HSBBAC - Backup Data Set File

To define the account codes to the Space Analyzer, you need to modify the sample member VCAACCT in `sharedprefix.MICS.PARMS`. VCAACCT defines the number of account codes that are to be carried in the above files and the length and the label associated with each account code. Its definitions apply to CA MICS database units that contain the Space Analyzer.

### PREPARATION

Each installation has its own method for associating data set allocations with the responsible user, projects, or departments. Before defining VCA account codes, it is important that you investigate your installation's accounting standards to:

- o Identify the current organizational coding system (cost center coding system identifying the division, department, project, and employee) and determine how the account codes are specified. For example, the division and department may be determined by a table lookup operation that associates the high-level qualifier of a DSNNAME with the owner's organizational codes.
- o Identify if, and how, the codes are verified to ensure that they correspond to a valid definition. We recommend that account code validation be performed in all cases and that unidentified or invalid account codes be assigned to a special installation overhead account code. This approach provides you with two benefits. First, it groups all invalid codes under the same category and therefore requires less DASD space for storing the VCA Information Area files. Second, it enables you to easily determine how much of

this overhead activity is taking place.

Consider the following suggestions in determining the number of account codes required to meet your installation's reporting and analysis needs:

- o Account codes are part of the summarization keys for the files that support them. As such, at least one record is generated for each combination of values. Therefore, the higher the number of account codes, the more DASD space the database will require. But at the same time, a higher number of account codes also supports data analysis at a greater detail.
- o If you anticipate needing to expand the account code structure in the future, establish an extra account code now to eliminate the need to retrofit the database later to add the new account code.

#### DEFINING ACCOUNT CODES

You can define up to nine levels of account codes to the Space Analyzer. The account codes are named VCAACT1 to VCAACT9 (and HSMACT1 to HSMACT2) in the CA MICS database.

Figure 7-1 provides a worksheet for defining the account codes. You can fill in the worksheet as you go through the following explanation on coding VCAACCT.

The VCAACCT member in sharedprefix.MICS.PARMS contains one statement per account code definition. Each statement contains several parameters, which are free-form (separated by one or more blanks) but positional. Blank statements are not permitted. Comments can be coded by beginning the statement with an asterisk (\*).

The statement format is:

```
level length 'descriptive title'  
  
or  
  
level mask length 'descriptive title'
```

where:

level =       The level of importance of each account code,  
                  with level 1 being the most important and the

highest level number being the least important. The levels are defined sequentially starting with 1. You can define up to nine levels. This parameter is required.

`mask =` An optional parameter that can be used to deactivate the account code in specified time-spans. Specifying a timespan mask requires coding `T(.....)` as the second parameter in a statement, where each "." represents a file timespan, in the order of `DETAIL`, `DAYS`, `WEEKS`, `MONTHS`, `YEARS`, and `TABLES`. For each timespan, you can specify a `Y` to indicate that the account code is active or an `N` to indicate that the account code is inactive.

If this parameter is skipped, it has a default value of `T(YYYYYY)`.

The following rules apply if you code the mask:

- o `DETAIL` must always be `Y`. Account codes may not be deactivated in this timespan.
- o If `DAYS` is `N`, then `WEEKS`, `MONTHS`, and `YEARS` must also be `N`. If the account code is inactive in the `DAYS` timespan, it cannot be active in higher timespans.
- o If `MONTHS` is `N`, then `YEARS` must be `N`. If the account code is inactive in the `MONTHS` timespan, it cannot be active in the `YEARS` timespan.

`length =` The length of the account code. The length may range from 1 to 30. This parameter is required.

`descriptive title =` The title that describes the account code. The length of the title is 1 to 40 characters. The title is embedded in quotes ('). This is a required parameter.

#### EXAMPLES

A sample `VCAACCT` with three account codes is shown below:

```
1 2 'DIVISION'
2 1 'REGION'
```

3 3 'USER'

A sample VCAACCT with three account codes, the last two of which are turned off in the MONTHS and YEARS timespans:

- 1                    2 'DIVISION'
- 2 T(YYNNN)        8 'DEPARTMENT'
- 3 T(YYNNN)        4 'USERID'

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: VCA Account Code Specifications |
| PARS Library Member is VCAACCT |
| Reference: Section 7.2.1, CA MICS Space Analyzer Guide |
+-----+
| ACCOUNT TIME-SPAN |
| CODE LEVEL MASK LENGTH DESCRIPTIVE TITLE |
| (1-9) |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
+-----+
| .....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+
    
```

Figure 7-1. VCA Account Code Specifications Worksheet

## 7.2.2 Account Code Derivation Exit (VCAACRT)

After you have defined VCAACCT for the number of account codes to be carried in the VCA files, your next step is to code the VCA Account Code Derivation Exit (VCAACRT) in sharedprefix.MICS.PARMS to create the SAS routine that will derive the account code data elements during the DAY090 step. For example, if you have defined four account codes in VCAACCT, then the VCAACRT exit must contain SAS code to assign the data elements VCAACT1, VCAACT2, VCAACT3, and VCAACT4 for each detail record processed.

```

*-----*
* IMPORTANT: If you intend to support the HSM      *
* component of the CA MICS Space Analyzer,        *
* please keep in mind that the objective is the   *
* creation of a single routine for your site that *
* is included in DAY090 processing for VCA and in *
* DAY095 processing for HSM. If a coding example  *
* shows VCAACRT techniques, it will be important *
* to keep in mind that the VCAACRT code will be  *
* included in HSM too and therefore the code     *
* should confine itself to common elements.      *
*-----*

```

You are responsible for testing the accuracy of the exit routine; some sample exit coding techniques are provided here to give you ideas for adapting your own routine. A worksheet for coding the VCAACRT exit is shown in Figure 7-2.

To determine the account code value, most sites use the data element DSNAME (data set name), which is kept in the DETAIL timespan.

For convenience, several "work" variables are assigned for use by your exit. The following example shows the SAS statements provided by VCA and HSM prior to calling the VCAACRT or HSMACRT exit respectively:

```

DSNODE1 = SCAN(DSNAME,1,' .');          /* ISOLATE */
DSNODE2 = SCAN(DSNAME,2,' .');          /* 1ST FIVE */
DSNODE3 = SCAN(DSNAME,3,' .');          /* NODES OF */
DSNODE4 = SCAN(DSNAME,4,' .');          /* DATA SET */
DSNODE5 = SCAN(DSNAME,5,' .');          /* NAME */
                                           /* LAST NODE*/
DSNODEL = LEFT(REVERSE(SCAN(REVERSE(DSNAME),1,' .')));

SKIP_REC = 0;          /* ALLOW VCAACRT TO SET SKIP_REC */
FILEID   = 'VCADAA';  /* ALLOW VCAACRT TO TEST FILEID */

```

```
EXIT='VCAACRT';
%INCLUDE SHRPARMS(VCAACRT);
EXIT=BLANKS;
/*-----*/
```

### Notes:

1. DSNODE1 through DSNODE5 are not permanent variables in either VCAGENIN or HSMGENIN. They are created simply for your coding convenience in accessing the qualifiers of the DSNAME of the current observation.
2. DSNODEL is the last node of the DSNAME.
3. SKIP\_REC can be set to 1 by your exit code to DELETE the current observation. This function is also available in the file exit associated with each file (for example, USRSDAA could set SKIP\_REC too).
4. FILEID is set by calling module to the following values to give your exit code the opportunity to know for which file it has been called to assign account codes:

```
FILEID = 'VCADAA';    /* When called for VCADAA file */
FILEID = 'VCA_VS';   /* When called for VCA_VS file */
FILEID = 'HSMIMG';   /* When called for HSMIMG file */
FILEID = 'HSBBAC';   /* When called for HSBBAC file */
```

Careful testing of this temporary variable by your VCAACRT routine allows your code to access file-specific variables if needed (REMEMBER - VCAACRT is %INCLUDEd by HSMACRT !).

The VCAACRT exit may contain any valid SAS statements for a data step except RETURN and DELETE. Coding either of these two statements may cause unpredictable results in DAILY processing, since the input processing is performed within a loop and control is not returned to the beginning of the DATA step for each record input.

### VCAACRT Coding Example 1

In the following example, assume that you have defined three account code levels in the VCAACCT PARMS member and that they are derived as follows:

VCAACT1 - is the first three characters of the high-level node of the data set name.

VCAACT2 - is the second three characters of the high-level node of the data set name.

VCAACT3 - is the second node of the data set name if it is three characters in length.

The standard data set naming convention is that the high-level node of the data set name is either three or six characters in length and the second-level node is three characters in length. An overhead category is defined for each account code level and the following special case is handled for data sets whose first account code is 'SYS'. If the high-level node of such a data set is not exactly three characters in length, the second- and third-level account codes are set to the overhead category.

The VCAACT member for the account definition would be:

```
1 3 'DIVISION'
2 3 'REGION'
3 3 'PROJECT'
```

The SAS code defined for the user account code exit would be:

```
/*-----*/
/*
/* SAMPLE VCA ACCOUNT CODE DERIVATION EXIT
/*
/* DIVISION IS BUILT FROM POSITIONS 1-3 OF NODE ONE OF
/* THE DATA SET NAME.
/*
/* PROJECT IS BUILT FROM POSITIONS 4-6 OF NODE ONE OF
/* THE DATA SET NAME.
/*
/* USER IS BUILT FROM POSITIONS 1-3 OF NODE TWO OF THE
/* DATA SET NAME.
/*
/*
/* DSNODE1-DSNODE5 ARE VARIABLES THAT CONTAIN THE
/* FIRST FIVE NODES OF THE DATA SET NAME
/* RESPECTIVELY.
/*
/*-----*/
```

```
IF INDEX(DSNODE1, ' ') LE 3 THEN VCAACT1 = '***';
ELSE VCAACT1 = SUBSTR(DSNODE1,1,3);
```

```
IF INDEX(DSNODE1, ' ') LT 6 THEN VCAACT2 = '***';
ELSE VCAACT2 = SUBSTR(DSNODE1,4,3);
```

```

IF INDEX(DSNODE2, ' ') NE 3 THEN VCAACT3 = '***';
ELSE VCAACT3 = DSNODE2;

IF VCAACT1 EQ 'SYS' AND DSNODE1 NE 'SYS' THEN DO;
  VCAACT2 = '***';
  VCAACT3 = '***';
END;
ACCTRTEX:          /* a label that the code could GOTO */

```

VCAACRT Coding Example 2

In the next example, a PROC FORMAT is used to build a single account code, VCAACT1.

The VCAACCT member for the account definition for this example would be:

```
1 8 'DIVISION'
```

The FORMAT used to define the data set name to account code mapping would be specified as follows:

```

PROC FORMAT PRINT ;

VALUE $ACTLIST
  'TSOGP1*'           = ACCTGRP1
  'TSOGP2*'           = ACCTGRP2
  'TSOGP3*'           = ACCTGRP3
  'TSOGP4*'           = ACCTGRP4
  'TSO.PRINT.DATA'    = ACCTPRNT
  'SYS1.VERY.LONG.DATASET.NAME' = ACCTPRNT
  .
  .
  .

OTHER                = ACCTOVRD ;

```

For those of you new to SAS, this SAS PROC FORMAT builds a table that when compiled forms a load-module. References to the value, \$ACTLIST, are resolved with a binary search technique by SAS.

The above format could then be used in the VCAACRT exit to map data set names to account codes as follows. Data sets beginning with TSO whose first level index has a length of six are mapped to their associated account groups. If a data set name does not conform to this naming convention, then it must have a unique entry in the table. Otherwise, it is

mapped to the overhead account.

The SAS code defined for the user account code exit would be as follows:

```

/*-----*/
/*
/* SAMPLE VCA ACCOUNT CODE DERIVATION EXIT USING FORMATS */
/*
/* IN THIS EXAMPLE ONLY ONE ACCOUNT CODE, DIVISION, */
/* IS DEFINED. */
/*
/* DIVISION IS BUILT USING THE $ACTLIST FORMAT WHICH WAS */
/* PREVIOUSLY BUILT USING PROC FORMAT. */
/*
/* $ACTLIST HAS DEFINED AN ACCOUNT CODE FOR ALL DATA SETS */
/* BELONGING TO THE TSO GROUPS. ALL DATA SETS OF THIS */
/* TYPE HAVE THE TSO ID AS THE FIRST LEVEL INDEX OF THE */
/* DATA SET NAME. ANY DATA SET THAT DOES NOT CONFORM */
/* TO THIS CONVENTION HAS A UNIQUE ENTRY IN THE $ACTLIST */
/* TABLE. ANY DATA SET NOT DEFINED IN THE TABLE IS */
/* MAPPED TO THE 'ACCTOVRD' ACCOUNT. */
/*
/*-----*/

```

```
IF (DSNAME= 'TSO') AND (INDEX(DSNAME, '.') EQ 7) THEN
```

```
    VCAACT1 = PUT(SUBSTR(DSNAME,1,6) || '*' , $ACTLIST.);
```

```
ELSE VCAACT1 = PUT(DSNAME, $ACTLIST);
```

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: VCA Account Code Routine Definition |
| PARS Library Member is VCAACRT |
| Reference: Section 7.2.2, CA MICS Space Analyzer Guide |
+-----+
| * VALIDATE FOR VALID ACCOUNT CODES, WHERE POSSIBLE ; |
|   IF account data is not valid GOTO ACCTOVHD ; |
|-----|
|-----|
|-----|
|-----|
| * BUILD ACCOUNT CODE FIELDS |
|   VCAACT1=field source 1 ; |
|   VCAACTn=field source n ; |
|-----|
|-----|
|-----|
|-----|
| GOTO ACCTRTEX ; |
| * LINKED ROUTINE TO BUILD INSTALLATION OVERHEAD ACCOUNT CODES ; |
| ACCTOVHD: |
|   VCAACT1='overhead category' ; |
|   VCAACTn='overhead category' ; |
|-----|
|-----|
|-----|
|-----|
| ACCTRTEX: |
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+

```

Figure 7-2. VCA Account Code Derivation Exit Worksheet

### 7.2.3 Synchronize HSMACCT with VCAACCT

Since the HSM and HSB information areas provide information about inactive and backup data sets, it should be evident that any methods of determining ownership and account structures for primary (active) data sets should apply in the same manner to data sets managed by DFHSM.

Data set ownership should not change simply because a data set is migrated due to lack of use -- a data set could be recalled to primary storage at any time. Similarly, the "owner" of a backup version should surely be the same "owner" as the active data set on primary storage.

In fact, to properly integrate the data organized in the VCA, HSM, and HSB information areas it is vital to populate the HSMACCTx data elements in the same manner as the VCAACTx elements. Therefore, they must have the same structure.

Modify sharedprefix.MICS.PARMS(HSMACCT) changing the distributed sample to mirror your current copy of VCAACCT in the same complex level library. As an example:

```
* HSM ACCOUNT CODE SPECIFICATION (SAME AS VCAACCT)
  1 10 'DIVISION'
  2  4 'SUMMARY1'
  3  8 'SUMMARY2'
  4  8 'DETAIL'
```

This modification to create equivalency between HSMACCT and VCAACCT must be done before HSMCGEN is run. A worksheet is shown below for your convenience but ISPF/PDF COPY or EDIT is the easiest tool for this task. Simply copy the content of sharedprefix.MICS.PARMS(VCAACCT) to replace HSMACCT in the same library.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: HSM Account Code Specifications |
| PARS Library Member is HSMACCT |
| Reference: Section 7.2.3, CA MICS Space Analyzer Guide |
+-----+
| ACCOUNT TIME-SPAN |
| CODE LEVEL MASK LENGTH DESCRIPTIVE TITLE |
| (1-9) |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
| - T(____) -- '_____' |
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+

```

Figure 7-3. HSM Account Code Specifications Worksheet

## 7.2.4 Synchronize HSMACRT with VCAACRT

The next member needing attention is HSMACRT in sharedprefix.MICS.PARMS. Like its counterpart, VCAACRT (discussed with examples in Section 7.2.2 of this guide), HSMACRT will assign the actual data values to the HSMACT1 through HSMACTn variables defined by the HSMACCT member discussed above.

Because HSMACRT is invoked as part of a SAS DATA step by CA MICS DAILY processing (step DAY095), it can exploit the %INCLUDE facility and therefore has been delivered as follows:

```
LENGTH %VCAACCT(OPTIONS=LEN);
%INCLUDE SHRPARMS(VCAACRT);
HSMACT1 = VCAACT1;
HSMACT2 = VCAACT2;
HSMACT3 = VCAACT3;
HSMACT4 = VCAACT4;
```

Note that the series of assignment statements above should reflect the definition of account structures from member HSMACCT. That is, if four account codes were defined, then four account codes must be assigned.

VCAACRT should normally confine itself to examining the content of CA MICS common elements - particularly DSNAME and its nodes. HSM will parse the DSNAME into temporary variables DSNODE1, DSNODE2, DSNODE3, DSNODE4, DSNODE5, and DSNODEL prior to calling the HSMACRT routine.

The HSMACRT routine is called when building both the HSMIMG and the HSBAC files. Therefore, it is important that your VCAACRT routine (which is being %INCLUDEd in the example) avoid any references to any VCA-specific elements. You may want to particularly watch out for any references to DAADSTYP or VOLSER.

If you must access file-specific variables, the temporary element FILEID should be tested to isolate these tests from the common part of the code that deals with DSNAME and its nodes.

As more and more installations implement volume pooling, ownership determined on the basis of VOLSER becomes more and more of an anachronism. As always, should you have questions about this vital area, please contact the CA MICS Product Support Group to discuss your implementation ideas.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: HSM Account Code Routine Definition |
| PARS Library Member is HSMACRT |
| Reference: Section 7.2.4, CA MICS Space Analyzer Guide |
+-----+
| * VALIDATE FOR VALID ACCOUNT CODES, WHERE POSSIBLE ; |
|   IF account data is not valid GOTO ACCTOVHD ; |
|-----|
|-----|
|-----|
|-----|
|-----|
| * BUILD ACCOUNT CODE FIELDS |
|   HSMACT1=field source 1 ; |
|   HSMACTn=field source n ; |
|-----|
|-----|
|-----|
|-----|
|-----|
| GOTO ACCTRTEX ; |
| * LINKED ROUTINE TO BUILD INSTALLATION OVERHEAD ACCOUNT CODES ; |
| ACCTOVHD: |
|   HSMACT1='overhead category' ; |
|   HSMACTn='overhead category' ; |
|-----|
|-----|
|-----|
|-----|
|-----|
|-----|
| ACCTRTEX: |
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+

```

Figure 7-4. HSM Account Code Derivation Exit Worksheet

### 7.2.5 Analyzer Definition Statements (VCAGENIN)

Each CA MICS component has a member that defines component generation statements in sharedprefix.MICS.GENLIB. The member's name is cccGENIN, where ccc is the three-character component identifier. The CA MICS Space Analyzer Option generation definition member is VCAGENIN.

Although each component is delivered with default statements, you are allowed to customize these statements based on your needs at your particular site. To activate your new definitions, you must submit sharedprefix.MICS.CNTL(VCACGEN) to complete the updates.

Chapter 4 of the CA MICS System Modification Guide describes the statements that comprise the GENIN members. Statements specific to this component are described below.

(Optional) The INPUTSOURCE Statement

-----

Lists the data source (SMF) and record types supported by the CA MICS Space Analyzer Option.

The default statement delivered with this component is:

```
INPUTSOURCE SMF 198
```

Indicates that the data source is SMF data, record type 198. Edit this statement if your site does not use this record type and submit sharedprefix.MICS.CNTL(VCACGEN) to activate your new definitions.

The OPTION Statement

-----

Note: An OPTION statement must be coded to specify the types of data you will provide to CA MICS.

Indicates which groups of data elements are to be kept in the database. Check the default definition and change it accordingly to meet your needs.

The default statement delivered with this component is:

```
OPTION NOEXTENTDETAIL HFS PDSEUSED/NOPDSEUS
```

It indicates that all elements that are cluster-coded for

this option will not be kept in the database. If you require EXTENTDETAIL elements, you must change the option from NOEXTENTDETAIL to EXTENTDETAIL and submit sharedprefix.MICS.CNTL(VCACGEN) to accept your updates.

The HFS/NOHFS option activates or inactivates the VCADIR file in the complex. If you generate data using HFSSCAN, you must ensure that this option is set to HFS.

Note: Because the CA MICS Space Analyzer Option utilizes directory names, you must consider the following:

DIRNAME elements 2 through 6 are delivered inactive. If you have directory entry names longer than 200 characters, you must ensure that DIRNAME2 to DIRNAME6 are active, if appropriate.

Also, if necessary, adjust the timespan indicators on the DIRNAME1 - DIRNAME6 NAME statements in sharedprefix.MICS.GENLIB(VCAGENIN).

Note: For more information about timespan indicators, see Chapter 6 of the System Modification Guide.

Optional statement PDSEUSED/NOPDSEUS is supported. It controls the way the used space values are assigned for PDSE and HFS data sets. The PDSEUSED keyword directs the DAILY processing routine to keep the PDSE and HFS data sets' used space value in the Data Set Allocation (VCADAA) file. These values are present by default in the VCC records produced by the CA MICS Space Collector Option.

The NOPDSEUS keyword is assumed when the optional keyword is missing. This default keyword directs the DAILY processing routine to back out the PDSE and HFS data sets' used space values in the Data Set Allocation (VCADAA) and Volume Allocation (VCAVOA) files.

## 7.3 Unit Level Parameters

Unit level parameters control the processing of the Space Analyzer within each database unit. These parameters are stored in the prefix.MICS.PARMS library associated with each database unit you define. The parameters are incorporated into the CA MICS system by the parameter generation process. Note that you must run the generation processes for these options to take effect.

Recommendation: Separate Units for VCA and HSM

While HSM could be added to the same unit as VCA, we suggest you strongly consider a separate unit for HSM. This is especially true if you manage a relatively large enterprise with many data sets under the ownership of DFHSM.

A separate unit for HSM will allow for the following:

- better testing and validation of this newer data source while leaving the VCA unit in production status uncluttered by the update activity of HSM.
- concurrent or overlapping execution of the DAILY job for HSM (step DAY095) with the DAILY job for other CA MICS units including VCA (step DAY090).
- a database split between the HSM (Migrate data) and the HSB (Backup data) information areas.

If you choose to add HSM to an existing database unit, follow Section 3.8.2.2 in the PIOM.

- When you arrive at Step 8 (PARMS definition) in the checklist for 3.8.2.2, it will point you back to this section for detailed instructions.
- When you arrive at Step 10 (DBMODEL definition) in the checklist for 3.8.2.2, it will point you back to this section for details.

If you choose to add HSM to a new database unit, follow Section 3.8.3.2 in the PIOM.

- When you arrive at Step 11 (PARMS content) in the checklist for this section of the PIOM, follow guidelines discussed in the next few sections of this document.

Figure 7-5 identifies each of the system code generation processes for the Space Analyzer, their PARMS member inputs, and the output MACROs (a member may contain the definition of more than one SAS MACRO) and FORMATs that they generate. The output libraries are at the database unit level (prefix) unless noted otherwise.

System Code Generator	MICS.PARMS Input Member	USER.SOURCE Output Members and SAS MACRO Names	USER.LOAD Output SAS FORMATs
VCAPGEN	VCAOPS	\$VCAMSTR VCADUR	(see VCAFMT)
HSMPGEN	HSMOPS	\$HSMSTR HSMDUR	n/a

Figure 7-5. cccPGEN Code Generator Cross-Reference List

The sections that follow show you how to specify the operational parameters that control how input data is processed by the Space Analyzer. The names in parentheses are the member names within the prefix.MICS.PARMS data set that contain the parameters described in the section.

- 1 - VCA Processing Options (VCAOPS)
- 2 - HSM Processing Options (HSMOPS)
- 3 - INPUTRDR, INPUTVCA, and INPUTHSM PARMS Members
- 4 - Database Space Modeling (DBMODEL)

### 7.3.1 VCA Processing Options (VCAOPS)

This section shows you how to specify the operational statements that control CA MICS Space Analyzer Option processing.

Operational statements are stored in the prefix.MICS.PARMS cccOPS member, where ccc is the component identifier, and are incorporated into the CA MICS system by running the prefix.MICS.CNTL(cccPGEN) job.

```
*****
*
* NOTE: CHANGES to prefix.MICS.PARMS(cccOPS) members *
* REQUIRE EXECUTION of prefix.MICS.CNTL(cccPGEN) *
* to take effect. *
*
* In addition, any change to parameters that *
* impact the DAILY operational job JCL such as, *
*
* o changing RESTART NO to RESTART YES, *
*
* o WORK parameter changes when RESTART NO is in *
* effect, *
*
* o Specifying TAPEfff (if this product supports *
* a DETAIL level TAPE option), *
*
* o or changes to prefix.MICS.PARMS(INPUTccc), *
*
* will require regeneration of the DAILY job by *
* executing prefix.MICS.CNTL(JCLGEN) or by *
* specifying DAILY in prefix.MICS.PARMS(JCLGEN) *
* and executing prefix.MICS.CNTL(JCLGEN). *
*
* Refer to the checklist (if provided) for updating *
* cccOPS parameters and running required generation *
* jobs. *
*****
```

Review the defaults provided in VCAOPS. In general, the defaults have been chosen to reduce the size of the CA MICS database. If the defaults meet your data center's requirements, you do not need to tailor VCAOPS.

#### General Syntax Rules

- o Comments are accepted and signified by an asterisk (\*) in

column 1.

- o Statement names and values can be entered in either upper case or lower case characters.
- o Statements can start anywhere in columns 1-72. Only one statement per card image is supported.
- o Statements can appear in any order. If you code multiple OPTIONS statements, they must be grouped together.

Statements

- o Description of the ACCOUNTING statement
- o Description of the BCSREQUIRED statement
- o Description of the EXTENTDETAIL statement
- o Description of the OPTIONS statement
- o Description of the VCAFORMAT statement
- o Description of the SUPDFSMS statement
- o Description of the WORK file options
- o Description of the Internal Step Restart options
- o Description of the Incremental Update options

ACCOUNTING Timespan  
-----

Valid values for timespan are DETAIL and DAYS. The default value is DAYS. This parameter specifies to VCA when it should invoke the accounting routines generated by CA MICS Accounting and Chargeback Option.

DAYS results in VCA presenting a summarized observation to the accounting routine. The sequence/summary elements of the VCA file at the DAYS timespan are used to create a summarized observation that represents all DASD space occupied by that control break.

DAYS allows reasonable flexibility in qualification pricing in the accounting routine. For example, assume the sequence/summary elements at the DAYS level are:

SYSID    VCAACT1    VCAACT2    DAADSTYP    DEVTYPE

STORSTGC STORMGTC YEAR MONTH DAY

CA MICS summarization routines will create a single observation for data sets whose values for these fields form a unique combination.

To build on this example, it follows that all data sets that have the following will be summarized into one observation:

```
SYSID='ASYS';  
VCAACT1='DIV100';  
VCAACT2='DEPT22';  
DAADSTYP='VS';  
DEVTYPE='3390-2';  
STORSTGC='NEVCACHE'  
STORMGTC='TESTDATA';  
YEAR=90;  
MONTH=07;  
DAY=24;
```

Accounting qualification techniques and algorithms allow for pricing of DASD space by applying rates based on the actual values in these control or sequence elements in the above example.

Note that VOLSER and DSNAME are absent from the list. If you need exception level pricing by having the accounting routines examine either VOLSER or DSNAME (or any other character data element not in the SORT key of the DAYS timespan), consider coding ACCOUNTING DETAIL here in VCAOPS.

DETAIL causes VCA to invoke the accounting routine to price a data set at the DETAIL timespan. Each data set can be examined by the accounting cost algorithms and all data elements carried at the DETAIL timespan are available for inspection by the accounting code.

To continue with the example above, VOLSER could be tested and a special rate could be applied to some DSNAMEs if they appeared on a given VOLSER.

The choice of DETAIL versus DAYS on this parameter should be worked out with the people responsible for the CA MICS Accounting and Chargeback Option. Since there will naturally be more observations at the DETAIL timespan than at the DAYS level, it follows that DETAIL will increase the space required for cycles of the accounting journal files (ACTJDA01, ACTJVS01, etc.). There is a one-for-one correspondence between the number of observations on the VCA

file and the accounting journal file that holds the charges developed from that VCA file.

BCSREQUIRED operand  
-----

Valid operands are YES and NO. The default is NO. The BCSREQUIRED option affects the manner in which observations for the VCA\_VS file (VSAM data sets) are constructed. Specify YES to require the BCS records for VSAM data sets or NO to allow observations for VSAM data sets to be written to the CA MICS database with BCS data missing.

Often, when excluding volumes during the Space Collector (VCC) run, a Base Catalog Structure (BCS) (user catalog) is not available for processing. When this happens, VCC only collects VVDS records describing the VSAM data sets.

The BCSREQUIRED parameter lets you specify that VSAM data set allocation information is or is not to be saved in the VCA\_VS file when matching BCS records are collected.

The VCC Processing Flags (DAAPFLAG) is set to indicate the presence or absence of the BCS record as follows:

'11.....'B - BCS and VVDS elements are present  
'01.....'B - BCS elements are not present

VCA processes the BCS to obtain the following data elements. When the BCS records are not present, some data elements are not valid in the VCA\_VS file. See Section 5.2.2.3 for a list of elements affected by this.

EXTENTDETAIL fff record  
-----

The EXTENTDETAIL statement has two operands:

fff is optional and is a file identifier. Valid values are DAA and \_VS. If fff is not specified, the value for record applies to both files.

record is required. Valid values are:  
YES - record details for extents  
NO - do not record details for extents

If EXTENTDETAIL is not coded, the default is as follows:

EXTENTDETAIL DAA NO  
EXTENTDETAIL \_VS NO

VSAM data sets that have more than 16 extents have one or more extra observations in the VCADAA and VCA\_VS DETAIL timespans that describe the 17th through 128th extents. Each extra observation describes the next set of 16 extents. The extra observations take DASD space and time to build and are important only if you run the TRACK MAP standard report or do very detailed extent reporting.

To save processing time and DASD space, you can specify that the extra observations are to be deleted during the DAILY DAY090 step. They can be deleted from either or both of the VCADAA and VCA\_VS files by coding the EXTENTDETAIL parameter accordingly.

For example, to keep all the details for extents 17 through 128 for VSAM on the VCA\_VS file but keep none on the VCADAA file, code two statements as follows:

```
EXTENTDETAIL _VS YES
EXTENTDETAIL DAA NO
```

To eliminate all recording of extent data beyond 16 extents in both files, code:

```
EXTENTDETAIL NO
```

Note: Saving space in the DETAIL timespan by coding EXTENTDETAIL NO makes the Volume Track Map Report unable to map the physical locations of the entire volume. Refer to Section 3.6 of this guide for an example of this report.

Because the Track Map Report uses the VCADAA file, you could save some space and still produce the report by coding:

```
EXTENTDETAIL _VS NO
EXTENTDETAIL DAA YES
```

Certain reports in the CA MICS StorageMate Option look at extent locations of the VTOC and the VVDS and are not affected adversely if you code EXTENTDETAIL NO.

```
$SCOL
OPTIONS sysid default_duration
-----
```

The OPTIONS statement is optional. If coded, both operands must be coded.

Valid values for sysid are 4-character SMF SYSIDs of VCA systems at your site or an asterisk, \*. The value of sysid

must match an SMF SYSID coded in prefix.MICS.PARMS(SYSID).

Valid values for default\_duration are integers, from 1 to 999, representing hours. This value is used to compute DASD storage occupancy for a data sets if VCC does not generate an indication that a particular data set was processed previously. For data sets previously processed, the value for default\_duration is computed by the VCA input format routine from the current and previous sample times recorded by VCC.

This operand allows you to specify a different default DURATION for each SYSID being processed by VCA. If your site allows VCA to accept data from any SMF SYSID and if the duration between VCA runs is a constant, use the statement format:

```
OPTIONS * nn
```

where \* means any SYSID and nn is the duration in hours.

If your site restricts the data coming into VCA or if the duration between VCC runs is not a constant, code individual OPTIONS statements for each SMF SYSID. OPTIONS statements must be grouped together.

The examples below illustrate valid and invalid VCAOPS coding:

Valid Example	Invalid Example
-----	-----
ACCOUNTING DETAIL	ACCOUNTING DETAIL
BCSREQUIRED YES	OPTIONS SYS1 12
OPTIONS SYS1 12	BCSREQUIRED YES
OPTIONS TST1 24	OPTIONS TST1 24
OPTIONS REMT 12	OPTIONS REMT 12

The example on the right causes improper code to be generated during the VCAPGEN process and unpredictable results during DAILY processing because the OPTIONS statements are not adjacent to each other.

```
VCAFMT  
-----
```

There are no operands for this parameter, which is optional. Adding VCAFMT to VCAOPS and executing VCAPGEN causes the following to be generated:

```
%INCLUDE SOURCE(VCAFMT);
```

This causes VCAPGEN to compile the PROC FORMATS that are distributed in sharedprefix.MICS.SOURCE(VCAFMT). You only need to do this if directed to by CA MICS maintenance instructions or if you have made a local modification to VCAFMT.

```
SUPDFSMS
```

```
-----
```

There are no operands for this parameter, which is optional. Adding SUPDFSMS to VCAOPS and executing VCAPGEN suppresses the VCA00602W and VCA00603W messages from printing in the MICSLOG. These messages are printed for data sets that are not SMS-managed but they reside on a volume that is SMS-managed.

```
WORK
```

```
----
```

This statement is optional. It enables sites experiencing either SAS WORK space allocation problems or out of work space conditions during DAYnnn or INCRnnn (where nnn is the job step number), daily or incremental update processing, to allocate multiple WORK files.

You can allocate multiple WORK files for use during the daily and/or incremental update job step. The maximum number of WORK files you can allocate varies by product. These additional work files are used in conjunction with the single work data set allocated by default using the JCLDEF parameters WORKUNIT and WORKSPACE.

Because the individual space allocation requirement for each WORK file is typically much smaller, it is more likely to be satisfied.

To take advantage of multiple WORK files support, edit prefix.MICS.PARMS(cccOPS) and insert a WORK statement as shown below:

```
WORK n data_set_allocation_parameters
```

where n is the number of WORK data sets

NOTE: The default is three (3).

The maximum is nine (9).

`data_set_allocation_parameters` is one or more data set allocation parameters (for example, `STORCLAS` or `SPACE`) separated by spaces.

You can also specify the `WORK` parameter as the following:

```
WORK n XXX pppp ssss
```

where:

`n` is the number of `WORK` data sets  
`XXX` is `TRK` or `CYL`  
`pppp` is the primary allocation  
`ssss` is the secondary allocation

Note: When allocating any number of SAS `WORK` data sets, be aware that one additional SAS `WORK` data set is automatically allocated to facilitate sorting. For example, if you allocate six SAS `WORK` data sets, you will actually get seven.

If you omit the `data_set_allocation_parameters` or the `WORK` parameter, the work data sets are allocated according to the values you specified for the `WORKUNIT` and `WORKSPACE` parameters in `prefix.MICS.PARMS(JCLDEF)`. Use the `data_set_allocation_parameters` to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the `WORK` data sets, `DAYnnn` and/or `INCRnnn` processing will fail and can only be restarted from the beginning.

Note: If internal step restart is active, you can override the `WORK` data set allocation parameters at execution-time using the `//PARMOVRD` facility. For more information about execution-time override of dynamic data set allocation parameters, see the `PIOM`, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS `LIBNAME` statement syntax. If you need multiple lines, repeat the `WORK` keyword on the continuation line.

`WORK` accepts the engine/host options documented in the SAS Companion for the z/OS environment, including `STORCLAS`, `UNIT`, `SPACE`, `BLKSIZE`, `DATACLAS`, `MGMTCLAS`, and `VOLSER`.

Important! Do not specify the DISP parameter.

Example 1:

```
WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss),RLSE)
```

where:

- n - is the number of WORK data sets.
- STORCLAS - specifies a storage class for a new data set.  
The name can have up to 8 characters.
- SPACE - specifies how much disk space to provide for  
a new data set being allocated.
- XXX - is TRK or CYL.
- pppp - is the primary allocation.
- ssss - is the secondary allocation.
- RLSE - specifies that free-space should be released  
when the data set is closed.

Example 2:

```
WORK n XXX pppp ssss
```

where:

- n - is the number of WORK data sets.
- XXX - is TRK or CYL.
- pppp - is the primary allocation.
- ssss - is the secondary allocation.

Example 3 (multiple lines):

```
WORK n STORCLAS=MICSTEMP UNIT=SYSDA  
WORK SPACE=(xxxx,(pppp,ssss),,,ROUND))
```

where:

- n - is the number of WORK data sets.
- STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.
- UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.
- SPACE - specifies how much disk space to provide for  
a new data set being allocated.
- XXX - is TRK or CYL.
- pppp - is the primary allocation.
- ssss - is the secondary allocation.

Note: Since there is some performance impact when using multiple WORK files, you should specify the minimum number of WORK data sets to meet your work space requirements. As a start, try incrementing the number gradually beginning from the default.

WORK Considerations  
-----

How Much Space Should You Allocate?

o First Time Implementation of Multiple Work Files

If this is the first time you are implementing multiple work files for this product in this unit, review prefix.MICS.PARMS(JCLDEF) and find the WORKSPACE parameter. It will resemble this sample statement:

```
WORKSPACE      TRK 500 250
```

The value shows the current SAS WORK space allocation for the unit as a single data set. It also serves as the default value used in the unit's DAYnnn daily update (and/or INCRnnn incremental update) step unless you provide a WORK parameter.

To achieve the equivalent work space allocation of WORKSPACE TRK 500 250 using multiple WORK data sets that will collectively share the work space requirements of the daily and/or incremental update step, you could code either one of these:

```
WORK 2 SPACE=(TRK,(250,125))
```

```
WORK 5 SPACE=(TRK,(100,50))
```

To determine the total work space, multiply the number of WORK files (n) by the primary (pppp) and secondary (ssss) values specified.

Note: To simplify the example, only the SPACE parameter is shown above. You can follow either with data set allocation parameters like UNIT or STORCLAS as required for your site.

o Adjusting Allocation for Existing Multiple WORK Files

If you have previously implemented multiple WORK file

support for this product in this unit, and you want to change either the number of WORK files or the space allocations, examine prefix.MICS.PARMS(cccOPS) and find the existing WORK statement.

- If the existing WORK statement only specifies the number of WORK files but does not contain space allocation information as shown below:

```
WORK 5
```

Then each of the multiple WORK files is allocated using the values from the WORKSPACE parameter of prefix.MICS.PARMS(JCLDEF), as described earlier under First Time Implementation of Multiple Work Files.

To increase workspace, you can increase the number of WORK files (for example, change WORK 5 to WORK 6,7,8, or 9), increase the space allocation in the WORKSPACE parameter, or do both.

To decrease workspace, you can decrease the number of WORK files (for example, change WORK 5 to WORK 4,3,2, or 1), decrease the space allocation in the WORKSPACE parameter, or do both.

You can also elect to explicitly specify the multiple WORK file space allocation by adding the space allocation values directly to the WORK statement. This will remove the link to the prefix.MICS.PARMS(JCLDEF) WORKSPACE parameter for multiple WORK file space allocation. This is recommended as it serves to clearly document, in one place, how multiple WORK files are allocated.

- If the existing WORK statement does include space allocation as shown in the examples below:

```
WORK 5 TRK 200 100
```

or

```
WORK 5 SPACE=(TRK,(200,100)) STORCLAS=MICSTEMP
```

Simply change the values to meet your needs.

If you need more work space, you can increase the number of WORK files (for example, change WORK 5 to WORK 6,7,8, or 9), increase the space allocation (for

example, change TRK 200 100 to TRK 250 120), or do both.

To decrease work space, you can decrease the number of WORK files (for example, change WORK 5 to WORK 4,3,2, or 1), decrease the space allocation (for example, change TRK 200 100 to TRK 150 80), or do both.

Note: If internal step restart is NOT active (RESTART NO) and you change the WORK parameter, you must:

- o Run cccPGEN
- o Run JCLGENU for DAILY (to regenerate DAILY) and, if incremental update is enabled, INCRccc

When internal step restart is active, (RESTART YES), then, when you change WORK and run cccPGEN, changes take effect immediately. There is no need to run JCLGENU.

### SASWORK

-----

This statement is optional.

The WORK DD statement in the CA MICS procedures allocates a temporary data set where SAS keeps its temporary data files and other items that SAS uses during processing of the current job.

By default, the space allocated is defined in the member prefix.MICS.PARMS(JCLDEF) with the WORKSPACE and WORKUNIT parameters, then generated into all the JCL procedures for a given unit.

With the SASWORK statement you have the option to override this unit-wide definition to specify the space allocation individually for the current step.

The format of the SASWORK statement is:

```
SASWORK data_set_allocation_parameters
```

where data\_set\_allocation\_parameters is one or more data set allocation parameters (for example, STORCLAS or SPACE) separated by spaces.

You can also specify the SASWORK parameter as the following:

```
SASWORK XXX pppp ssss
```

where:

```
XXX is TRK or CYL
pppp is the primary allocation
ssss is the secondary allocation
```

If you omit the `data_set_allocation_parameters` or the `SASWORK` statement, the `WORK` data set is allocated according to the values you specified for the `WORKUNIT` and `WORKSPACE` parameters in `prefix.MICS.PARMS(JCLDEF)`. Use the `data_set_allocation_parameters` to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the `SASWORK` keyword on the continuation line.

Example:

```
SASWORK STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))
```

where:

```
STORCLAS - specifies a storage class for a new data set.
           The name can have up to 8 characters.
SPACE     - specifies how much disk space to provide for
           a new data set being allocated.
XXX       - is TRK or CYL.
pppp      - is the primary allocation.
ssss      - is the secondary allocation.
```

Note: If you change the `SASWORK` parameter, you must:

- o Run `cccPGEN`
- o Run `JCLGENU` for `DAILY` (to regenerate `DAILY`) and, if incremental update is enabled, `INCRccc`

```
MULTWORK|NOMULT fff fff ... fff
```

```
-----
```

Since multiple work files usage impacts performance, this

product provides these optional parameters so you can restrict multiple work files usage to only those files having excessive space requirements.

Note: You can only use one of these optional parameters with the WORK statement, NOT both.

The MULTWORK parameter restricts the use of multiple WORK files to ONLY those listed after the MULTWORK keyword.

```
MULTWORK fff fff ... fff
```

where fff is the unique three character identifier

If you need multiple lines, repeat the MULTWORK on the continuation line.

The NOMULT parameter forces the use of multiple WORK files for all files EXCEPT those specified after the NOMULT keyword.

```
NOMULT fff fff ... fff
```

where fff is the unique three character identifier

If you need multiple lines, repeat the NOMULT on the continuation line.

The default is as follows if neither MULTWORK nor NOMULT parameters are specified:

```
MULTWORK _VS _VT BCS BCT DAW EDA EVS EVT NVR SMS TAW VOA
```

The following files are eligible for multiple WORK support:

_VS	VSAM DATA SET ALLOCATION FILE (<16 EXTENTS)
_VT	Internal Work File Parallel to _VS File
BCS	BCS DATA SET ALLOCATION FILE
BCT	Internal Work File Parallel to BCS File
DAW	DATA SET ALLOCATION FILE (<16 EXTENTS)
EDA	DATA SET ALLOCATION FILE (>16 EXTENTS)
EVS	VSAM DATA SET ALLOC. FILE (>16 EXTENTS)
EVT	Internal Work File Parallel to EVS File
NVR	NON-VSAM RECORDS FILE
SMS	STORAGE MANAGEMENT SYSTEM DATASET FILE
TAW	Internal Work File Parallel to DAW File
VOA	VOLUME ALLOCATION FILE
DIR	USS Directory Entry File
DIW	USS Directory Entry Work File (Update Phase)

FIL USS File System Work File  
 FIW USS File System Work File (Update Phase)

RESTART YES/NO  
 -----

This statement is optional. Specify this to activate internal step restart for this product's DAILY and/or INCRccc database update job steps:

RESTART YES

If you do not specify or enable the RESTART parameter, then this option defaults to the following and internal step restart is disabled:

RESTART NO

```
*****
*                                                                 *
* Note: Changing the RESTART parameter (either from NO        *
*       to YES or from YES to NO) requires regeneration      *
*       of the DAILY operational job by executing            *
*       prefix.MICS.CNTL(JCLGEN) or by specifying            *
*       DAILY in prefix.MICS.PARMS(JCLGEN) and                *
*       executing prefix.MICS.CNTL(JCLGEN).                    *
*                                                                 *
*       If incremental update is active for this product,    *
*       you must also regenerate the INCRccc job.             *
*                                                                 *
*****
```

Internal step restart can significantly reduce time and resource usage to recover from daily and/or incremental update processing failures. CA MICS uses a checkpoint/restart technique.

- o When internal step restart is activated, the database update job step "checkpoints" (or saves) intermediate results (work file contents) and the operational environment at the end of each processing phase.
- o Then, if required, the database update step can resume execution at the beginning of the processing phase in which the failure occurred.

- o Restart is accomplished by restoring the operational environment from the last checkpoint, bypassing completed processing phases, and resuming execution using intermediate results (work files) from the last checkpoint.

Note: When you activate internal step restart (RESTART YES), the following optional restart parameters are enabled. These parameters have no effect if restart is disabled (RESTART NO). For more details, see the individual parameter descriptions later in this section.

- o RESTARTCKPT data\_set\_allocation\_parameters
- o RESTARTWORK data\_set\_allocation\_parameters
- o DYNAMWAIT minutes

Processing Phases:  
-----

This product employs three database update processing phases followed by the two common roll-up phases.

Phase	Description
-----	-----
FORMAT	Read raw input data, convert to SAS format, and output to intermediate work files.
SORT	Sort intermediate work file contents, eliminate duplicate input data, and prepare for DETAIL cycle creation.
DBUPDATE	Merge data across optional multiple work files, enhance data content, and create the new DETAIL cycle.
DYSUM	Summarize DETAIL data to create new DAYS cycles and to update current week-to-date and month-to-date cycles.
DYAGE	Cutover new database cycles to production and "age" existing cycles.

---

## RESTART Considerations

-----

### o Overhead

Enabling internal step restart adds some overhead to the database update job step -- the cost of taking checkpoints and managing saved materials. Since this overhead is relatively constant and independent of input data volume, you may find that costs outweigh potential savings when input data volume is low, for example in a test unit. For high volume, production units, internal step restart support overhead should be a minor portion of total resource usage.

### o Cataloged Work Files

When internal step restart is enabled, the SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged with permanent dataset names so they will be retained for use in restart if the step abends. These data sets are deleted when the step completes successfully.

Prior to enabling internal step restart support, these data sets were probably allocated on system "scratch" space with a temporary, system assigned data set names. If your installation standards do not allow "permanent" data sets on DASD volumes used for temporary work space, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct the internal step restart data sets to a generic unit or storage class that allows cataloged data sets.

### o Dynamic Allocation

When internal step restart is active, dynamic allocation is employed for the work data sets. If your installation restricts dynamic allocation of large, cataloged data sets, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct work data set allocation to a generic unit or storage class where dynamic allocation is allowed.

### o Data Set Names

The SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged according to the standard CA MICS unit database

data set name conventions. The default DDNAME and data set names are:

- o SAS work data set,  
//cccXWORK DD DSN=prefix.MICS.cccXWORK,.....
- o Internal step restart control data set,  
//cccXCKPT DD DSN=prefix.MICS.cccXCKPT,.....
- o Multiple work file data sets,  
//WORKnn DD DSN=prefix.MICS.cccWRKnn,.....

Since these data sets conform to the same data set name conventions as your existing CA MICS data sets, there should be few, if any, data set name related allocation issues. However, it is possible to override the data set names if required. Please contact CA MICS Product Support for assistance if you must alter data set names.

RESTARTCKPT  
-----

This statement is optional. Specify the following to override default data set allocation parameters for the internal step restart checkpoint data set:

RESTARTCKPT data\_set\_allocation\_parameters

Note: RESTARTCKPT is ignored when you specify RESTART NO.

The internal step restart checkpoint data set (or cccXCKPT data set) contains processing status, control, and SAS environmental information for internal step restart processing checkpoints. This includes a copy of the SAS WORK format and macro catalogs, current macro variable values, and a description of work files that may be needed to restart DAYnnn processing.

By default, the cccXCKPT data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Specify RESTARTCKPT to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the cccXCKPT

data set, DAYnnn processing will fail and can only be restarted from the beginning.

Note: You can override the RESTARTCKPT data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the RESTARTCKPT keyword on the continuation line.

RESTARTCKPT accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
RESTARTCKPT STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
pp is the primary allocation  
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
RESTARTCKPT STORCLAS=MICSTEMP UNIT=SYSDA
RESTARTCKPT SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.

- UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.
  
- SPACE - specifies how much disk space to provide for  
a new data set being allocated.

RESTARTWORK

-----

This statement is optional. Specify the following to override default data set allocation parameters for the internal step restart WORK data set:

RESTARTWORK data\_set\_allocation\_parameters

Note: RESTARTWORK is ignored when you specify RESTART NO.

The internal step restart WORK data set (or cccXWORK data set) contains the intermediate work files that are not enabled to multiple work file support, including those files you may have specified on the optional NOMULT statement.

By default, the cccXWORK data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Specify RESTARTWORK to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the cccXWORK data set, DAYnnn processing will fail and can only be restarted from the beginning.

Note: You can override the RESTARTWORK data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the RESTARTWORK keyword on the continuation line.

RESTARTWORK accepts the engine/host options documented in "SAS Companion for the z/OS Environment", including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
RESTARTWORK STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for  
a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
pp is the primary allocation  
ss is the secondary allocation

and ROUND specifies that the allocated space be  
"rounded" to a cylinder boundary when the unit  
specified was a block length. ROUND is ignored  
with the TRK or CYL options.

Example 2 (multiple lines):

```
RESTARTWORK STORCLAS=MICSTEMP UNIT=SYSDA
RESTARTWORK SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
The name can have up to 8 characters.

SPACE - specifies how much disk space to provide for  
a new data set being allocated.

INCRUPDATE

-----

This statement is optional. Specify this to enable  
incremental update for this product:

INCRUPDATE YES

If you do not specify or enable the INCRUPDATE parameter, then this option defaults to this and incremental update is disabled:

INCRUPDATE NO

```
*****
*
* Note: Changing the INCRUPDATE parameter (either from NO
*       to YES or from YES to NO) requires regeneration
*       of the DAILY operational job by executing
*       prefix.MICS.CNTL(JCLGEN) or by specifying
*       DAILY in prefix.MICS.PARMS(JCLGEN) and
*       executing prefix.MICS.CNTL(JCLGEN).
*
*       If you specify INCRUPDATE YES, you must also
*       generate the INCRccc, cccIUALC, and cccIUGDG jobs
*       (where ccc is the 3 character product ID).
*       Depending on the options you select, you may also
*       need to execute the cccIUALC and/or cccIUGDG
*       jobs.
*
*****
```

Incremental update can significantly reduce time and resource usage in the DAILY job by letting you split out a major portion of daily database update processing into multiple, smaller, incremental updates executed throughout the day.

- o Standard CA MICS database update processing involves (1) reading and processing raw input data to generate DETAIL and DAYS level CA MICS database files, followed by (2) summarization of DETAIL/DAYS level data to update week-to-date and month-to-date database files.
- o When you activate incremental update:
  - You can execute the first-stage processing (raw data input to create DETAIL/DAYS files) multiple times throughout the day, each time processing a subset of the total day's input data.
  - Then, during the final update of the day (in the DAILY job), the incremental DETAIL/DAYS files are "rolled-up" to the database DETAIL and DAYS timespans, and then summarized to update the week-to-date and month-to-date files.
- o Incremental update is independent of your internal step

restart or DBSPLIT specifications. You have the option to perform incremental updates with or without internal step restart support.

- o Incremental update is activated and operates independently by product. The incremental update job for this product, INCRccc (where ccc is the product ID), can execute concurrently with the incremental update job for another product in the same unit database.
- o The CA MICS database remains available for reporting and analysis during INCRccc job execution.

```
*****
*
* Note: CA MICS is a highly configurable system
* supporting up to 36 unit databases, each of which
* can be configured and updated independently.
* Incremental update is just one of the options you
* can use to configure your CA MICS complex.
*
* All efforts should be made to employ CA MICS
* configuration capabilities to minimize issues
* prior to activating incremental update. For
* example:
*
* o Splitting work to multiple units is an
* effective way to enable parallel database
* update processing
*
* o Adjusting account code definitions to ensure
* adequate data granularity while minimizing
* total database space and processing time
*
* o Tailoring the database to drop measurements
* and metrics of lesser value to your
* data center, thereby reducing database update
* processing and resource consumption
*
* While incremental update is intended to reduce
* DAILY job elapsed time, total resource usage of
* the combined INCRccc and DAILY jobs steps can
* increase due to the additional processing
* required to maintain the incremental update
* "to-date" files and for roll-up to the unit
* database. The increased total resource usage
* will be more noticeable with small data volumes,
* where processing code compile time is a greater
* percentage of total processing cost.
*
```

\* \*  
\*\*\*\*\*

Note: When you activate incremental update (INCRUPDATE YES), the following optional incremental update parameters are enabled. These parameters have no effect if incremental update is disabled (INCRUPDATE NO). For more details, see the individual parameter descriptions later in this section.

- o INCRDB PERM/TAPE/DYNAM
- o INCRDETAIL data\_set\_allocation\_parameters
- o INCRDAYS data\_set\_allocation\_parameters
- o INCRCKPT data\_set\_allocation\_parameters
- o INCRSPLIT USE/IGNORE data\_set\_allocation\_parameters

Incremental update processing reads and processes raw measurement data to create and maintain DETAIL and DAYS level "to-date" files for the current day.

- o These incremental update database files are maintained on unique z/OS data sets, independent of the standard CA MICS database files, and independent of any other product's incremental update database files. There is one data set each for DETAIL and DAYS level "to-date" data and a single incremental update checkpoint data set for this product in this unit.
- o The incremental update DETAIL and DAYS files can be permanent DASD data sets, or they can be allocated dynamically as needed and deleted after DAILY job processing completes. Optionally, you can keep the incremental update DETAIL and DAYS files on tape, with the data being loaded onto temporary DASD space as needed for incremental update or DAILY job processing. See the INCRDB PERM/TAPE/DYNAM option for more information.

After activating incremental update, you will use three incremental update facility jobs found in prefix.MICS.CNTL (Note that ccc is the product ID):

- o cccIUALC

You execute this job to allocate and initialize the incremental update checkpoint file, and optionally the

incremental update DETAIL and DAYS database files.  
cccIUALC is generally executed just ONE time.

o cccIUGDG

You execute this job to add generation data group (GDG) index definitions to your system catalog in support of the INCRDB TAPE option. cccIUGDG is generally executed just ONE time.

o INCRccc

This is the job you execute for each incremental update. You will integrate this job into your database update procedures for execution one or more times per day to process portions of the total day's measurement data.

Note: The DAILY job is run once at the end of the day. It will perform the final incremental update for the day's data, and then roll-up the incremental DETAIL/DAYS files to the database DETAIL and DAYS timespans and update the week-to-date and month-to-date files.

INCRUPDATE Considerations

-----

o Overhead

Incremental update is intended to reduce DAILY job resource consumption and elapsed time by offloading a major portion of database update processing to one or more executions of the INCRccc job. In meeting this objective, incremental update adds processing in the INCRccc and DAILY jobs to accumulate data from each incremental update execution into the composite "to-date" DETAIL and DAYS incremental update files, and also adds processing in the DAILY job to copy the incremental update files to the unit database DETAIL and DAYS timespans. The amount of this overhead and the savings in the DAILY job are site-dependent, and will vary based on input data volume and on the number of times INCRccc is executed each day.

In addition, activating incremental update will cause additional compile-based CPU time to be consumed in the DAYnnn DAILY job step. The increase in compile time is due to additional code included for each file structure in support of the feature. This increase should be static

based on the scope of the CA MICS data integration product in terms of files. This compile-time increase does not imply an increase in elapsed or execution time.

Incremental update allows I/O bound, intensive processing (raw data inputting, initial CA MICS transformation, etc.) to be distributed outside of the DAILY job. I/O processing is the largest contributor to elapsed time in large volume applications. Thus, the expected overall impact is a decrease in the actual runtime of the DAYnnn job step.

- o Increased "Prime Time" Workload

By offloading work from the DAILY job to one or more INCRccc executions throughout the day, you are potentially moving system workload and DASD work space usage from the "off-hours," (when the DAILY job is normally executed) to periods of the day where your system resources are in highest demand. You should schedule INCRccc executions carefully to avoid adverse impact to batch or online workloads. For example, if your site's "prime shift" is 8:00 AM to 5:00 PM, you might choose to schedule incremental updates for 7:00 AM (just before "prime shift") and 6:00 PM (just after "prime shift"), with the DAILY job executing just after midnight.

- o Increased DASD Usage

The DASD space required for the incremental update DETAIL and DAYS database files is in addition to the DASD space already reserved for the CA MICS database. By default, the incremental update database files are permanently allocated, making this DASD space unavailable for other applications. In general, you can assume that the incremental update database files will require space equivalent to two cycles of this product's DETAIL and DAYS timespan files.

Alternatively, the incremental update database files can be allocated in the first incremental update of the day and deleted by the DAILY job (see the INCRDB DYNAM option later in this section). This approach reduces the amount of time that the DASD space is dedicated to incremental update, and lets the amount of DASD space consumed increase through the day as you execute each incremental update.

A third option is to store the incremental update database files on tape (see the INCRDB TAPE option).

With this approach, the DASD space is required just for the time that each incremental update or DAILY job step is executing. Note that while this alternative reduces the "permanent" DASD space requirement, the total amount of DASD space required while the incremental update or DAILY jobs are executing is unchanged. In addition, the TAPE option adds processing to copy the incremental update files to tape, and to reload the files from tape to disk.

Note: The incremental update checkpoint file is always a permanently allocated disk data set. This is a small data set and should not be an issue.

- o Operational Complexity

Incremental update expands your measurement data management and job scheduling issues. You must ensure that each incremental update and the DAILY job processes your measurement data chronologically; that is, each job must see data that is newer than the data processed by the prior job. By incrementally updating the database, you have more opportunities to miss a log file, or to process a log out of order.

- o Interval End Effects

Each incremental update processes a subset of the day's measurement data, taking advantage of early availability of some of the day's data, for example, when a measurement log fills and switches to a new volume. This can cause a problem if the measurement log split occurs while the data source is logging records for the end of a measurement interval, thus splitting the data for a single measurement interval across two log files. When an incremental update processes the first log file, the checkpoint high end timestamp is set to indicate that this split measurement interval has been processed. Then, when the rest of the measurement interval's data is encountered in a later update, it can be dropped as duplicate data (because data for this measurement interval end timestamp has already been processed).

Appropriate scheduling of log dumps and incremental updates can avoid this problem. For example, if you plan to run incremental updates at 7:00 AM and 6:00 PM, you could force a log dump in the middle of the measurement interval just prior to the scheduled incremental update executions. This is an extension of the procedure you

may already be using for end-of-day measurement log processing. The objective is to ensure that all records for each monitor interval are processed in the same incremental update.

o Dynamic Allocation

When you activate incremental update and specify TAPE or DYNAM for the INCRDB parameter, dynamic allocation is employed for the incremental update database files. If your site restricts dynamic allocation of large, cataloged data sets, you must use the INCRDETAIL and INCRDAYS parameters to direct incremental update data set allocation to a generic unit or storage class where dynamic allocation is allowed.

o Data Set Names

The incremental update database files are allocated and cataloged according to standard CA MICS unit database data set name conventions. The DDNAME and default data set names are (where ccc is the product ID):

- o Incremental update checkpoint file,  
//IUCKPT DD DSN=prefix.MICS.ccc.IUCKPT,.....
- o Incremental update DETAIL  
//IUDETAIL DD DSN=prefix.MICS.ccc.IUDETAIL,.....
- o Incremental update DAYS  
//IUDAYS DD DSN=prefix.MICS.ccc.IUDAYS,....

Since these data sets conform to the same data set name conventions as your existing CA MICS data sets, there should be few, if any, data-set-name-related allocation issues. However, it is possible to override the data set names if required. Contact Technical Support at <http://ca.com/support> for assistance if you must change data set names.

INCRDB  
-----

This statement is optional. The default is this:

INCRDB PERM

Note: INCRDB is ignored when you specify INCRUPDATE NO.

Specify this statement or take the default, to keep the incremental update database DETAIL and DAYS files on permanently allocated DASD data sets:

INCRDB PERM

Execute the prefix.MICS.CNTL(cccIUALC) job to allocate the incremental update database files.

```
*****
*
* Note: The incremental update checkpoint file is always *
*       a permanently allocated DASD data set.           *
*
*****
```

Specify this to offload the incremental update DETAIL and DAYS files to tape between incremental update executions:

INCRDB TAPE #gdgs UNIT=name

With the TAPE option, the incremental update DETAIL and DAYS DASD data sets are dynamically allocated at the beginning of the incremental update job or DAILY job step, and then are deleted after the job step completes.

- o The first incremental update job of the day allocates and initializes the incremental update database files. At the end of the job, the DETAIL and DAYS files are copied to a new (+1) generation of the incremental update tape data sets. Then the DASD files are deleted.
- o Subsequent incremental update jobs restore the DASD incremental update database files from the current, (0) generation, incremental update tape data sets before processing the input measurement data. At the end of the job, the DETAIL and DAYS files are copied to a new (+1) generation of the incremental update tape data sets. Then the DASD files are deleted.
- o The DAILY job step also restores the DASD incremental update database files from the (0) generation tape files before processing the input data, but does NOT copy the incremental update database files to tape. Thus, the DAILY job actually creates a new, null (+1) generation.
- o Use the #gdgs parameter to specify the maximum number of

incremental update tape generations. The minimum is 2 and the maximum is 99, with a default of 5. Set the number of generations equal to or greater than the number of incremental updates, including the DAILY job you plan to execute each day. This facilitates restart and recovery if you encounter problems requiring you to reprocess portions of the daily measurement data.

- o Use the optional UNIT=name parameter to specify a tape unit name for the incremental update database output tapes. The default is to use the same tape unit as the input tapes.
- o A special index must be created in your system catalog for each of the incremental update tape data set generation data groups. The prefix.MICS.CNTL(cccIUGDG) job will generate the statements to create the incremental update GDG index definitions.
  - Before each index is built, it is deleted. These DLTX (or DELETE) statements causes an error message if no entry exists. This is done so that you can change the number of entries without having to delete each of the index entries.
  - DLTX and BLDG (or DELETE and DEFINE) fail if there is a cataloged data set with the same index. IDCAMS (or IEHPRGM) issues a message and gives a return code of 8. This issue is not a problem for non-GDG entries or if the GDG already has the desired number of entries.
  - If you want to change the number of entries kept in a GDG with cataloged data sets, do the following:
    1. Uncatalog any existing entries in the GDG.
    2. Delete the index with a DLTX (or DELETE).
    3. Create the index with a BLDG (or DEFINE).
    4. Catalog any entries that are uncataloged in step 1.
- o The incremental update tape data set names are as follows, where ccc is the product ID:
  - Incremental update tape DETAIL file  
tapeprefix.MICS.ccc.IUXTAPE.GnnnnV00
  - Incremental update tape DAYS file  
tapeprefix.MICS.ccc.IUDTAPE.GnnnnV00

```

*****
*
* Note: The INCRDETAIL and INCRDAYS parameters are
*       required when you specify INCRDB TAPE.
*
*****

```

Specify this parameter to allocate dynamically the incremental update DETAIL and DAYS DASD data sets in the first incremental update of the day, and then delete these data sets at the end of the DAILY job step:

INCRDB DYNAM

- o With this option, no space is used for the incremental update database files during the time between the end of the DAILY job step and the beginning of the next day's first incremental update.
- o With this approach, you can set the data set allocation parameters so that the incremental update DETAIL and DAYS data sets start out with a minimum allocation and then grow through secondary allocations as more space is required for subsequent incremental updates. For example, enough space for one incremental update.

```

*****
*
* Note: The INCRDETAIL and INCRDAYS parameters are
*       required when you specify INCRDB DYNAM.
*
*****

```

INCRCKPT

-----

This statement is optional. Specify this to override default data set allocation parameters for the incremental update checkpoint data set:

INCRCKPT        data\_set\_allocation\_parameters

Note: INCRCKPT is ignored when you specify INCRUPDATE NO.

The incremental update checkpoint data set tracks incremental update job status and the data that has been processed during the current daily update cycle. The incremental update checkpoint is used to detect and block the input of duplicate data during incremental update processing. This data set will be exactly the same size as prefix.MICS.CHECKPT.DATA (the unit checkpoint data set), usually 20K to 200K depending on the prefix.MICS.PARMS(SITE) CKPTCNT parameter (100-9999).

Your INCRCKPT parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update checkpoint file. If you specified INCRDB PERM, then the cccIUALC job will also allocate the incremental update DETAIL and DAYS database files.
- o By default the incremental update checkpoint data set is allocated as SPACE=(TRK,(5,2)) using the value you specified for the prefix.MICS.PARMS(JCLDEF) DASDUNIT parameter.
- o Omit the INCRCKPT parameter if you prefer to override data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRCKPT keyword on the continuation line.

INCRCKPT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
INCRCKPT STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.

The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
pp is the primary allocation  
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRCKPT  STORCLAS=MICSTEMP UNIT=SYSDA
INCRCKPT  SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

INCRDAYS

-----

This statement is required if you specify either of these:

INCRDB TAPE

INCRDB DYNAM

Otherwise, this statement is optional. There is no default.

Specify this to define data set allocation parameters for the incremental update DAYS data set (IUDAYS):

INCRDAYS      data\_set\_allocation\_parameters

Note: INCRDAYS is ignored when you specify INCRUPDATE NO.

The incremental update DAYS data set (IUDAYS) contains the current incremental update days-level database files, and the DAYS "to-date" data for the current daily update cycle. You should allocate DASD space equivalent to two cycles of this product's DAYS timespan data.

If you specified INCRDB PERM (the default), your INCRDAYS parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update database and checkpoint files.
- o Omit the INCRDAYS parameter if you prefer to specify data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

If you specified INCRDB TAPE or INCRDB DYNAM, your INCRDAYS parameter specifications are used in incremental update DAYS data set dynamic allocation during incremental update or DAILY job step execution.

- o The INCRDAYS parameter is required for the TAPE or DYNAM option.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRDAYS keyword on the continuation line.
- o INCRDAYS accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

- o You can override the INCRDAYS data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, Section 2.3.6.

Example 1:

```
INCRDAYS    STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for  
a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
pp is the primary allocation  
ss is the secondary allocation

and ROUND specifies that the allocated space be  
"rounded" to a cylinder boundary when the unit  
specified was a block length. ROUND is ignored  
with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRDAYS    STORCLAS=MICSTEMP UNIT=SYSDA
INCRDAYS    SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for  
a new data set being allocated.

```
INCRCKPT
-----
```

This statement is optional. Specify this to override default  
data set allocation parameters for the incremental update  
checkpoint data set:

```
INCRCKPT    data_set_allocation_parameters
```

Note: INCRCKPT is ignored when you specify INCRUPDATE NO.

The incremental update checkpoint data set tracks incremental update job status and the data that has been processed during the current daily update cycle. The incremental update checkpoint is used to detect and block the input of duplicate data during incremental update processing. This data set will be exactly the same size as prefix.MICS.CHECKPT.DATA (the unit checkpoint data set), usually 20K to 200K depending on the prefix.MICS.PARMS(SITE) CKPTCNT parameter (100-9999).

Your INCRCKPT parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update checkpoint file. If you specified INCRDB PERM, then the cccIUALC job will also allocate the incremental update DETAIL and DAYS database files.
- o By default the incremental update checkpoint data set is allocated as SPACE=(TRK,(5,2)) using the value you specified for the prefix.MICS.PARMS(JCLDEF) DASDUNIT parameter.
- o Omit the INCRCKPT parameter if you prefer to override data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRCKPT keyword on the continuation line.

INCRCKPT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
INCRCKPT STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.

The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
pp is the primary allocation  
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRCKPT  STORCLAS=MICSTEMP UNIT=SYSDA
INCRCKPT  SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

INCRSPLIT  
-----

This statement is optional and defaults to this:

INCRSPLIT IGNORE

Specify the following if you want the incremental update job for this product to get input measurement data from the output of the SPLITSMF job. The optional data\_set\_allocation\_parameters are used by the SPLITSMF job when creating the measurement data file for this product.

INCRSPLIT USE data\_set\_allocation\_parameters

Note: INCRSPLIT is ignored when you specify INCRUPDATE NO.

This option would be used when multiple products in a single unit database are enabled to incremental update. The SPLITSMF job performs the same function for incremental update jobs as the DAILY job DAYSMF step performs for the DAYnnn database update steps.

- o The SPLITSMF job dynamically allocates, catalogs, and populates prefix.MICS.ccc.IUSPLTDS data sets for each product in the unit database for which you specified both the INCRUPDATE YES and INCRSPLIT USE parameters. These data sets are then deleted after processing by the appropriate INCRccc job.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRSPLIT keyword on each continuation line.
- o INCRSPLIT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Specify the following or accept the default if you want the incremental update jobs for this product to get their input measurement data from the data sets specified in the INPUTccc (or INPUTSMF) member of prefix.MICS.PARMS:

INCRSPLIT IGNORE

When you specify INCRSPLIT IGNORE, this product will NOT participate in SPLITSMF job processing.

Example 1:

```
INCRSPLIT USE STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.

The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
pp is the primary allocation  
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRSPLIT USE STORCLAS=MICSTEMP UNIT=SYSDA
INCRSPLIT     SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

DYNAMWAIT

-----

This statement is optional. Specify the following:

DYNAMWAIT minutes

to override the default amount of time, in minutes, the DAILY and/or INCRccc job will wait for an unavailable data set.

Note: This optional parameter is not normally specified.  
The system default is adequate for most data centers.

Internal Step Restart and Incremental Update facilities use

z/OS dynamic allocation services to create new data sets and to access existing data sets. Data set naming conventions and internal program structure are designed to minimize data set contention. However, if data set allocation does fail because another batch job or online user is already using a data set, DAILY and/or INCRccc processing will wait 15 seconds and then try the allocation again. By default, the allocation will be attempted every 15 seconds for up to 15 minutes. After 15 minutes, the DAILY or INCRccc job will abort.

If data set contention in your data center does cause frequent DAILY or INCRccc job failures, and you are unable to resolve the contention through scheduling changes, you may want to use the DYNAMWAIT parameter to increase the maximum number of minutes the DAILY and/or INCRccc jobs will wait for the data set to become available.

On the other hand, if your data center standards require that the DAILY and/or INCRccc jobs fail immediately if required data sets are unavailable, specify the following:

```
DYNAMWAIT 0
```

Note: You can override the DYNAMWAIT parameter at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: VCA Options Definition |
| PARM5 Library Member is VCAOPS |
| Reference: Section 7.3.1 |
+-----+
| VCA PROCESSING OPTIONS: |
| ACCOUNTING _____ (DETAIL or DAYS) |
| |
| BCSREQUIRED _____ (YES or NO) |
| EXTENTDETAIL _____ (optional: DAA, _VS, or blank for both) |
| _____ (YES or NO) |
| OPTIONS _____ (sysid) |
| _____ (default duration) |
| VCAFMT _____ (no operand) |
| WORK _____ (optional: n data_set_allocation_parameters) |
| RESTART _____ (optional: YES or NO) |
| INCRUPDATE _____ (optional: YES or NO) |
| INCRDB _____ (optional: PERM, TAPE, or DYNAM) |
| INCRDETAIL _____ (optional: data_set_allocation_parameters) |
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70... |
+-----+

```

Figure 7-6. VCA Options Definition Worksheet

### 7.3.2 HSM Processing Options (HSMOPS)

HSMOPS has been patterned after VCAOPS. While it has fewer parameters than VCAOPS, the syntax and functionality are the same.

This section shows you how to specify the operational statements that control CA MICS Space Analyzer processing for HSM component.

Operational statements are stored in the prefix.MICS.PARMS cccOPS member, where ccc is the component identifier, and are incorporated into the CA MICS system by running the prefix.MICS.CNTL(cccPGEN) job.

```
*****
*
* NOTE: CHANGES to prefix.MICS.PARMS(cccOPS) members *
* REQUIRE EXECUTION of prefix.MICS.CNTL(cccPGEN) *
* to take effect. *
*
* In addition, any change to parameters that *
* impact the DAILY operational job JCL such as, *
*
* o changing RESTART NO to RESTART YES, *
*
* o WORK parameter changes when RESTART NO is in *
* effect, *
*
* o Specifying TAPEfff (if this product supports *
* a DETAIL level TAPE option), *
*
* o or changes to prefix.MICS.PARMS(INPUTccc), *
*
* will require regeneration of the DAILY job by *
* executing prefix.MICS.CNTL(JCLGEND) or by *
* specifying DAILY in prefix.MICS.PARMS(JCLGENU) *
* and executing prefix.MICS.CNTL(JCLGENU). *
*
* Refer to the checklist (if provided) for updating *
* cccOPS parameters and running required generation *
* jobs. *
*****
```

Review the defaults provided in HSMOPS. In general, the defaults have been chosen to reduce the size of the CA MICS database. If the defaults meet your installation's requirements, you do not need to tailor HSMOPS.

#### General Syntax Rules

- o Comments are accepted and signified by an asterisk (\*) as the first non-blank character.
- o Statement names and values can be entered in either upper case or lower case characters.
- o Statements can start anywhere in columns 1-72. Only one statement per card image is supported.
- o Statements can appear in any order. If you code multiple OPTIONS statements, they must be grouped together.

#### Statements

- o Description of the ACCOUNTING statement
- o Description of the OPTIONS statement
- o Description of the WORK files options
- o Description of the Internal Step Restart options
- o Description of the Incremental Update options

#### ACCOUNTING timespan

-----

Valid values for timespan are DETAIL and DAYS. The default value is DAYS. This parameter specifies to the HSM component (DAY095 step) when it should invoke the accounting routines generated by CA MICS Accounting and Chargeback.

DAYS results in CA MICS (DAY095 step) presenting an observation summarized to the DAYS sequence to the CA MICS costing routine. DSNAME will NOT be available at this summary level, but DAYS allows reasonable flexibility in qualification pricing in the accounting routine.

If you need access to the DSNAME variable, you should specify DETAIL for this parameter.

ACCOUNTING specifications for VCAOPS and HSMOPS are not required to match. For example, if you have a need to access DSNAME for the VCADAA file you would probably want similar

access to DSNAME for the HSMIG file. In this case, VCAOPS and HSMOPS would both specify "ACCOUNTING DETAIL".

The choice of ACCOUNTING DETAIL or ACCOUNTING DAYS applies to both the HSMIG file and the HSBAC file.

```
OPTIONS sysid default_duration
-----
```

The OPTIONS statement is optional; if coded, both operands must be coded.

Valid values for sysid are 4-character SMF SYSIDs of HSM systems at your site or an asterisk (\*). The value of sysid must match an SMF SYSID coded in prefix.MICS.PARMS(SYSID).

Valid values for default\_duration are integers from 1 to 999, representing hours. This value is used for the very first inventory of DFHSM resources performed by VCC. After this first data collection, all records produced for HSM by VCC will contain the "current" data collection time and the previous data collection time. Hence, the actual DURATION is computed by CA MICS and this default duration will be ignored. Since HSM data collection is usually once per day, this parameter is less important in HSMOPS than its corresponding usage in VCAOPS. In VCA, the default duration will be used when a new VOLSER is placed into service and there has never been a previous scan time for that volume.

The default is "OPTIONS \* 24" where \* means any SYSID and 24 is the duration in hours. While the default choice is highly recommended, managing multiple data centers and MVS images requires flexibility so the following discussion elaborates on syntactical details.

If your site restricts the data coming into VCA or if the duration between VCC runs is not a constant, code individual OPTIONS statements for each SMF SYSID. OPTIONS statements must be grouped together.

The examples below illustrate valid and invalid VCAOPS coding:

Valid Example:

```
ACCOUNTING DETAIL
OPTIONS SYS1 12
OPTIONS TST1 24
```

Invalid Example:

```
OPTIONS TST1 24
ACCOUNTING DETAIL
* COMMENT IS OK HERE
```

---

OPTIONS REMT 12

OPTIONS REMT 12

The example on the right causes improper code to be generated during the HSMPGEN process and unpredictable results during DAILY processing because the OPTIONS statements are not adjacent to each other.

WORK

----

This statement is optional. It enables sites experiencing either SAS WORK space allocation problems or out of work space conditions during DAYnnn or INCRnnn (where nnn is the job step number), daily or incremental update processing, to allocate multiple WORK files.

You can allocate multiple WORK files for use during the daily and/or incremental update job step. The maximum number of WORK files you can allocate varies by product. These additional work files are used in conjunction with the single work data set allocated by default using the JCLDEF parameters WORKUNIT and WORKSPACE.

Because the individual space allocation requirement for each WORK file is typically much smaller, it is more likely to be satisfied.

To take advantage of multiple WORK files support, edit prefix.MICS.PARMS(cccOPS) and insert a WORK statement as shown below:

```
WORK n data_set_allocation_parameters
```

where n is the number of WORK data sets

NOTE: The default is one (1).  
The maximum is nine (9).

data\_set\_allocation\_parameters is one or more data set allocation parameters (for example, STORCLAS or SPACE) separated by spaces.

You can also specify the WORK parameter as the following:

```
WORK n XXX pppp ssss
```

where:

n is the number of WORK data sets  
XXX is TRK or CYL  
pppp is the primary allocation  
ssss is the secondary allocation

Note: When allocating any number of SAS WORK data sets, be aware that one additional SAS WORK data set is automatically allocated to facilitate sorting. For example, if you allocate six SAS WORK data sets, you will actually get seven.

If you omit the `data_set_allocation_parameters` or the `WORK` parameter, the work data sets are allocated according to the values you specified for the `WORKUNIT` and `WORKSPACE` parameters in `prefix.MICS.PARMS(JCLDEF)`. Use the `data_set_allocation_parameters` to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the WORK data sets, `DAYnnn` and/or `INCRnnn` processing will fail and can only be restarted from the beginning.

Note: If internal step restart is active, you can override the WORK data set allocation parameters at execution-time using the `//PARMOVRD` facility. For more information about execution-time override of dynamic data set allocation parameters, see the `PIOM`, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS `LIBNAME` statement syntax. If you need multiple lines, repeat the `WORK` keyword on the continuation line.

`WORK` accepts the engine/host options documented in the SAS Companion for the z/OS environment, including `STORCLAS`, `UNIT`, `SPACE`, `BLKSIZE`, `DATACLAS`, `MGMTCLAS`, and `VOLSER`.

Important! Do not specify the `DISP` parameter.

Example 1:

```
WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss),RLSE)
```

where:

n - is the number of WORK data sets.  
STORCLAS - specifies a storage class for a new data set.

The name can have up to 8 characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

XXX - is TRK or CYL.

pppp - is the primary allocation.

ssss - is the secondary allocation.

RLSE - specifies that free-space should be released when the data set is closed.

Example 2:

```
WORK n XXX pppp ssss
```

where:

n - is the number of WORK data sets.

XXX - is TRK or CYL.

pppp - is the primary allocation.

ssss - is the secondary allocation.

Example 3 (multiple lines):

```
WORK n STORCLAS=MICSTEMP UNIT=SYSDA
WORK SPACE=(xxxx,(pppp,ssss),,,ROUND))
```

where:

n - is the number of WORK data sets.

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

XXX - is TRK or CYL.

pppp - is the primary allocation.

ssss - is the secondary allocation.

Note: Since there is some performance impact when using multiple WORK files, you should specify the minimum number of WORK data sets to meet your work space requirements. As a start, try incrementing the number gradually beginning from the default.

WORK Considerations  
-----

How Much Space Should You Allocate?

o First Time Implementation of Multiple Work Files

If this is the first time you are implementing multiple work files for this product in this unit, review prefix.MICS.PARMS(JCLDEF) and find the WORKSPACE parameter. It will resemble this sample statement:

```
WORKSPACE      TRK 500 250
```

The value shows the current SAS WORK space allocation for the unit as a single data set. It also serves as the default value used in the unit's DAYnnn daily update (and/or INCRnnn incremental update) step unless you provide a WORK parameter.

To achieve the equivalent work space allocation of WORKSPACE TRK 500 250 using multiple WORK data sets that will collectively share the work space requirements of the daily and/or incremental update step, you could code either one of these:

```
WORK 2 SPACE=(TRK,(250,125))
```

```
WORK 5 SPACE=(TRK,(100,50))
```

To determine the total work space, multiply the number of WORK files (n) by the primary (pppp) and secondary (ssss) values specified.

Note: To simplify the example, only the SPACE parameter is shown above. You can follow either with data set allocation parameters like UNIT or STORCLAS as required for your site.

o Adjusting Allocation for Existing Multiple WORK Files

If you have previously implemented multiple WORK file support for this product in this unit, and you want to change either the number of WORK files or the space allocations, examine prefix.MICS.PARMS(cccOPS) and find the existing WORK statement.

- If the existing WORK statement only specifies the number of WORK files but does not contain space allocation information as shown below:

```
WORK 5
```

Then each of the multiple WORK files is allocated using the values from the WORKSPACE parameter of prefix.MICS.PARMS(JCLDEF), as described earlier under First Time Implementation of Multiple Work Files.

To increase workspace, you can increase the number of WORK files (for example, change WORK 5 to WORK 6,7,8, or 9), increase the space allocation in the WORKSPACE parameter, or do both.

To decrease workspace, you can decrease the number of WORK files (for example, change WORK 5 to WORK 4,3,2, or 1), decrease the space allocation in the WORKSPACE parameter, or do both.

You can also elect to explicitly specify the multiple WORK file space allocation by adding the space allocation values directly to the WORK statement. This will remove the link to the prefix.MICS.PARMS(JCLDEF) WORKSPACE parameter for multiple WORK file space allocation. This is recommended as it serves to clearly document, in one place, how multiple WORK files are allocated.

- If the existing WORK statement does include space allocation as shown in the examples below:

```
WORK 5 TRK 200 100
```

or

```
WORK 5 SPACE=(TRK,(200,100)) STORCLAS=MICSTEMP
```

Simply change the values to meet your needs.

If you need more work space, you can increase the number of WORK files (for example, change WORK 5 to WORK 6,7,8, or 9), increase the space allocation (for example, change TRK 200 100 to TRK 250 120), or do both.

To decrease work space, you can decrease the number of WORK files (for example, change WORK 5 to WORK 4,3,2, or 1), decrease the space allocation (for example, change TRK 200 100 to TRK 150 80), or do both.

Note: If internal step restart is NOT active (RESTART NO) and you change the WORK parameter, you must:

- o Run cccPGEN
- o Run JCLGENU for DAILY (to regenerate DAILY) and, if incremental update is enabled, INCRccc

When internal step restart is active, (RESTART YES), then, when you change WORK and run cccPGEN, changes take effect immediately. There is no need to run JCLGENU.

### SASWORK

-----

This statement is optional.

The WORK DD statement in the CA MICS procedures allocates a temporary data set where SAS keeps its temporary data files and other items that SAS uses during processing of the current job.

By default, the space allocated is defined in the member prefix.MICS.PARMS(JCLDEF) with the WORKSPACE and WORKUNIT parameters, then generated into all the JCL procedures for a given unit.

With the SASWORK statement you have the option to override this unit-wide definition to specify the space allocation individually for the current step.

The format of the SASWORK statement is:

```
SASWORK data_set_allocation_parameters
```

where data\_set\_allocation\_parameters is one or more data set allocation parameters (for example, STORCLAS or SPACE) separated by spaces.

You can also specify the SASWORK parameter as the following:

```
SASWORK XXX pppp ssss
```

where:

```
XXX is TRK or CYL  
pppp is the primary allocation  
ssss is the secondary allocation
```

If you omit the data\_set\_allocation\_parameters or the SASWORK statement, the WORK data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE

parameters in prefix.MICS.PARMS(JCLDEF). Use the `data_set_allocation_parameters` to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the SASWORK keyword on the continuation line.

Example:

```
SASWORK STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))
```

where:

```
STORCLAS - specifies a storage class for a new data set.
           The name can have up to 8 characters.
SPACE     - specifies how much disk space to provide for
           a new data set being allocated.
XXX       - is TRK or CYL.
pppp     - is the primary allocation.
ssss     - is the secondary allocation.
```

Note: If you change the SASWORK parameter, you must:

- o Run cccPGEN
- o Run JCLGENU for DAILY (to regenerate DAILY) and, if incremental update is enabled, INCRccc

```
MULTWORK|NOMULT fff fff ... fff
-----
```

Since multiple work files usage impacts performance, this product provides these optional parameters so you can restrict multiple work files usage to only those files having excessive space requirements.

Note: You can only use one of these optional parameters with the WORK statement, NOT both.

The MULTWORK parameter restricts the use of multiple WORK files to ONLY those listed after the MULTWORK keyword.

```
MULTWORK fff fff ... fff
```

where fff is the unique three character identifier

If you need multiple lines, repeat the MULTWORK on the continuation line.

The NOMULT parameter forces the use of multiple WORK files for all files EXCEPT those specified after the NOMULT keyword.

NOMULT fff fff ... fff

where fff is the unique three character identifier

If you need multiple lines, repeat the NOMULT on the continuation line.

The default is

MULTWORK BAC BAX MIG MIX VOL VOX

if neither MULTWORK or NOMULT parameters are specified.

The following files are eligible for multiple WORK support:

HSM DFHSM Inventory Primary Information Area

MIG Migrated Data File  
MIX Internal Work File Parallel to MIG File  
TAP Tape Planning File  
TAX Internal Work File Parallel to TAP File  
VOL DASD Volume Planning File  
VOX Internal Work File Parallel to VOL File

HSB DFHSM Inventory Backup Information Area

BAC BACKUP Data File  
BAX Internal Work File Parallel to BAC File

RESTART YES/NO

-----

This statement is optional. Specify this to activate internal step restart for this product's DAILY and/or INCRccc database update job steps:

RESTART YES

If you do not specify or enable the RESTART parameter, then this option defaults to the following and internal step restart is disabled:

RESTART NO

```
*****
*
* Note: Changing the RESTART parameter (either from NO
*       to YES or from YES to NO) requires regeneration
*       of the DAILY operational job by executing
*       prefix.MICS.CNTL(JCLGEN) or by specifying
*       DAILY in prefix.MICS.PARMS(JCLGEN) and
*       executing prefix.MICS.CNTL(JCLGEN).
*
*       If incremental update is active for this product,
*       you must also regenerate the INCRccc job.
*
*****
```

Internal step restart can significantly reduce time and resource usage to recover from daily and/or incremental update processing failures. CA MICS uses a checkpoint/restart technique.

- o When internal step restart is activated, the database update job step "checkpoints" (or saves) intermediate results (work file contents) and the operational environment at the end of each processing phase.
- o Then, if required, the database update step can resume execution at the beginning of the processing phase in which the failure occurred.
- o Restart is accomplished by restoring the operational environment from the last checkpoint, bypassing completed processing phases, and resuming execution using intermediate results (work files) from the last checkpoint.

Note: When you activate internal step restart (RESTART YES), the following optional restart parameters are enabled. These parameters have no effect if restart is disabled (RESTART NO). For more details, see the individual parameter descriptions later in this section.

- o RESTARTCKPT data\_set\_allocation\_parameters

- o RESTARTWORK data\_set\_allocation\_parameters
- o DYNAMWAIT minutes

Processing Phases:

-----

This product employs two database update processing phases followed by the two common roll-up phases.

Phase	Description
-----	-----
FORMAT	Read raw input data, convert to SAS format, and output to intermediate work files.
DBUPDATE	Sort intermediate work file contents, eliminate duplicate input data, merge data across optional multiple work files, enhance data content, and create the new DETAIL cycle.
DYSUM	Summarize DETAIL data to create new DAYS cycles and to update current week-to-date and month-to-date cycles.
DYAGE	Cutover new database cycles to production and "age" existing cycles.

RESTART Considerations

-----

- o Overhead

Enabling internal step restart adds some overhead to the database update job step -- the cost of taking checkpoints and managing saved materials. Since this overhead is relatively constant and independent of input data volume, you may find that costs outweigh potential savings when input data volume is low, for example in a test unit. For high volume, production units, internal step restart support overhead should be a minor portion of total resource usage.

- o Cataloged Work Files

When internal step restart is enabled, the SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged with permanent dataset names so they will be retained for use in restart if the step abends. These data sets are deleted when the step completes successfully.

Prior to enabling internal step restart support, these data sets were probably allocated on system "scratch" space with a temporary, system assigned data set names. If your installation standards do not allow "permanent" data sets on DASD volumes used for temporary work space, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct the internal step restart data sets to a generic unit or storage class that allows cataloged data sets.

#### o Dynamic Allocation

When internal step restart is active, dynamic allocation is employed for the work data sets. If your installation restricts dynamic allocation of large, cataloged data sets, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct work data set allocation to a generic unit or storage class where dynamic allocation is allowed.

#### o Data Set Names

The SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged according to the standard CA MICS unit database data set name conventions. The default DDNAME and data set names are:

- o SAS work data set,  
//cccXWORK DD DSN=prefix.MICS.cccXWORK,.....
- o Internal step restart control data set,  
//cccXCKPT DD DSN=prefix.MICS.cccXCKPT,.....
- o Multiple work file data sets,  
//WORKnn DD DSN=prefix.MICS.cccWRKnn,.....

Since these data sets conform to the same data set name conventions as your existing CA MICS data sets, there should be few, if any, data set name related allocation issues. However, it is possible to override the data set

names if required. Please contact CA MICS Product Support for assistance if you must alter data set names.

RESTARTCKPT  
-----

This statement is optional. Specify the following to override default data set allocation parameters for the internal step restart checkpoint data set:

RESTARTCKPT data\_set\_allocation\_parameters

Note: RESTARTCKPT is ignored when you specify RESTART NO.

The internal step restart checkpoint data set (or cccXCKPT data set) contains processing status, control, and SAS environmental information for internal step restart processing checkpoints. This includes a copy of the SAS WORK format and macro catalogs, current macro variable values, and a description of work files that may be needed to restart DAYnnn processing.

By default, the cccXCKPT data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Specify RESTARTCKPT to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the cccXCKPT data set, DAYnnn processing will fail and can only be restarted from the beginning.

Note: You can override the RESTARTCKPT data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the RESTARTCKPT keyword on the continuation line.

RESTARTCKPT accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
RESTARTCKPT  STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for  
a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
pp is the primary allocation  
ss is the secondary allocation

and ROUND specifies that the allocated space be  
"rounded" to a cylinder boundary when the unit  
specified was a block length. ROUND is ignored  
with the TRK or CYL options.

Example 2 (multiple lines):

```
RESTARTCKPT  STORCLAS=MICSTEMP UNIT=SYSDA
RESTARTCKPT  SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for  
a new data set being allocated.

```
RESTARTWORK
```

```
-----
```

This statement is optional. Specify the following to  
override default data set allocation parameters for the  
internal step restart WORK data set:

RESTARTWORK data\_set\_allocation\_parameters

Note: RESTARTWORK is ignored when you specify RESTART NO.

The internal step restart WORK data set (or cccXWORK data set) contains the intermediate work files that are not enabled to multiple work file support, including those files you may have specified on the optional NOMULT statement.

By default, the cccXWORK data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Specify RESTARTWORK to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the cccXWORK data set, DAYnnn processing will fail and can only be restarted from the beginning.

Note: You can override the RESTARTWORK data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the RESTARTWORK keyword on the continuation line.

RESTARTWORK accepts the engine/host options documented in "SAS Companion for the z/OS Environment", including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
RESTARTWORK STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen

pp is the primary allocation  
 ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
RESTARTWORK STORCLAS=MICSTEMP UNIT=SYSDA
RESTARTWORK SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
 The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
 The name can have up to 8 characters.

SPACE - specifies how much disk space to provide for  
 a new data set being allocated.

INCRUPDATE

-----

This statement is optional. Specify this to enable incremental update for this product:

INCRUPDATE YES

If you do not specify or enable the INCRUPDATE parameter, then this option defaults to this and incremental update is disabled:

INCRUPDATE NO

```
*****
*
* Note: Changing the INCRUPDATE parameter (either from NO
*       to YES or from YES to NO) requires regeneration
*       of the DAILY operational job by executing
*       prefix.MICS.CNTL(JCLGEN) or by specifying
*       DAILY in prefix.MICS.PARMS(JCLGEN) and
*       executing prefix.MICS.CNTL(JCLGEN).
*
```

```
*           If you specify INCRUPDATE YES, you must also           *
*           generate the INCRccc, cccIUALC, and cccIUGDG jobs      *
*           (where ccc is the 3 character product ID).            *
*           Depending on the options you select, you may also     *
*           need to execute the cccIUALC and/or cccIUGDG          *
*           jobs.                                                 *
*                                                                 *
*****
```

Incremental update can significantly reduce time and resource usage in the DAILY job by letting you split out a major portion of daily database update processing into multiple, smaller, incremental updates executed throughout the day.

- o Standard CA MICS database update processing involves (1) reading and processing raw input data to generate DETAIL and DAYS level CA MICS database files, followed by (2) summarization of DETAIL/DAYS level data to update week-to-date and month-to-date database files.
- o When you activate incremental update:
  - You can execute the first-stage processing (raw data input to create DETAIL/DAYS files) multiple times throughout the day, each time processing a subset of the total day's input data.
  - Then, during the final update of the day (in the DAILY job), the incremental DETAIL/DAYS files are "rolled-up" to the database DETAIL and DAYS timespans, and then summarized to update the week-to-date and month-to-date files.
- o Incremental update is independent of your internal step restart or DBSPLIT specifications. You have the option to perform incremental updates with or without internal step restart support.
- o Incremental update is activated and operates independently by product. The incremental update job for this product, INCRccc (where ccc is the product ID), can execute concurrently with the incremental update job for another product in the same unit database.
- o The CA MICS database remains available for reporting and analysis during INCRccc job execution.

```
*****
*                                                                 *
```

```

* Note: CA MICS is a highly configurable system          *
* supporting up to 36 unit databases, each of which    *
* can be configured and updated independently.        *
* Incremental update is just one of the options you   *
* can use to configure your CA MICS complex.         *
*                                                     *
* All efforts should be made to employ CA MICS       *
* configuration capabilities to minimize issues       *
* prior to activating incremental update. For        *
* example:                                           *
*                                                     *
* o Splitting work to multiple units is an          *
* effective way to enable parallel database         *
* update processing                                 *
*                                                     *
* o Adjusting account code definitions to ensure    *
* adequate data granularity while minimizing        *
* total database space and processing time         *
*                                                     *
* o Tailoring the database to drop measurements     *
* and metrics of lesser value to your              *
* data center, thereby reducing database update    *
* processing and resource consumption              *
*                                                     *
* While incremental update is intended to reduce    *
* DAILY job elapsed time, total resource usage of   *
* the combined INCRccc and DAILY jobs steps can    *
* increase due to the additional processing         *
* required to maintain the incremental update      *
* "to-date" files and for roll-up to the unit     *
* database. The increased total resource usage     *
* will be more noticeable with small data volumes, *
* where processing code compile time is a greater  *
* percentage of total processing cost.             *
*                                                     *
*****

```

Note: When you activate incremental update (INCRUPDATE YES), the following optional incremental update parameters are enabled. These parameters have no effect if incremental update is disabled (INCRUPDATE NO). For more details, see the individual parameter descriptions later in this section.

- o INCRDB      PERM/TAPE/DYNAM
- o INCRDETAIL data\_set\_allocation\_parameters
- o INCRDAYS    data\_set\_allocation\_parameters

- o INCRCKPT data\_set\_allocation\_parameters
- o INCRSPLIT USE/IGNORE data\_set\_allocation\_parameters

Incremental update processing reads and processes raw measurement data to create and maintain DETAIL and DAYS level "to-date" files for the current day.

- o These incremental update database files are maintained on unique z/OS data sets, independent of the standard CA MICS database files, and independent of any other product's incremental update database files. There is one data set each for DETAIL and DAYS level "to-date" data and a single incremental update checkpoint data set for this product in this unit.
- o The incremental update DETAIL and DAYS files can be permanent DASD data sets, or they can be allocated dynamically as needed and deleted after DAILY job processing completes. Optionally, you can keep the incremental update DETAIL and DAYS files on tape, with the data being loaded onto temporary DASD space as needed for incremental update or DAILY job processing. See the INCRDB PERM/TAPE/DYNAM option for more information.

After activating incremental update, you will use three incremental update facility jobs found in prefix.MICS.CNTL (Note that ccc is the product ID):

- o cccIUALC

You execute this job to allocate and initialize the incremental update checkpoint file, and optionally the incremental update DETAIL and DAYS database files. cccIUALC is generally executed just ONE time.

- o cccIUGDG

You execute this job to add generation data group (GDG) index definitions to your system catalog in support of the INCRDB TAPE option. cccIUGDG is generally executed just ONE time.

- o INCRccc

This is the job you execute for each incremental update. You will integrate this job into your database update procedures for execution one or more times per day

to process portions of the total day's measurement data.

Note: The DAILY job is run once at the end of the day. It will perform the final incremental update for the day's data, and then roll-up the incremental DETAIL/DAYS files to the database DETAIL and DAYS timespans and update the week-to-date and month-to-date files.

#### INCRUPDATE Considerations

-----

##### o Overhead

Incremental update is intended to reduce DAILY job resource consumption and elapsed time by offloading a major portion of database update processing to one or more executions of the INCRccc job. In meeting this objective, incremental update adds processing in the INCRccc and DAILY jobs to accumulate data from each incremental update execution into the composite "to-date" DETAIL and DAYS incremental update files, and also adds processing in the DAILY job to copy the incremental update files to the unit database DETAIL and DAYS timespans. The amount of this overhead and the savings in the DAILY job are site-dependent, and will vary based on input data volume and on the number of times INCRccc is executed each day.

In addition, activating incremental update will cause additional compile-based CPU time to be consumed in the DAYnnn DAILY job step. The increase in compile time is due to additional code included for each file structure in support of the feature. This increase should be static based on the scope of the CA MICS data integration product in terms of files. This compile-time increase does not imply an increase in elapsed or execution time. Incremental update allows I/O bound, intensive processing (raw data inputting, initial CA MICS transformation, etc.) to be distributed outside of the DAILY job. I/O processing is the largest contributor to elapsed time in large volume applications. Thus, the expected overall impact is a decrease in the actual runtime of the DAYnnn job step.

##### o Increased "Prime Time" Workload

By offloading work from the DAILY job to one or more INCRccc executions throughout the day, you are

potentially moving system workload and DASD work space usage from the "off-hours," (when the DAILY job is normally executed) to periods of the day where your system resources are in highest demand. You should schedule INCRccc executions carefully to avoid adverse impact to batch or online workloads. For example, if your site's "prime shift" is 8:00 AM to 5:00 PM, you might choose to schedule incremental updates for 7:00 AM (just before "prime shift") and 6:00 PM (just after "prime shift"), with the DAILY job executing just after midnight.

- o Increased DASD Usage

The DASD space required for the incremental update DETAIL and DAYS database files is in addition to the DASD space already reserved for the CA MICS database. By default, the incremental update database files are permanently allocated, making this DASD space unavailable for other applications. In general, you can assume that the incremental update database files will require space equivalent to two cycles of this product's DETAIL and DAYS timespan files.

Alternatively, the incremental update database files can be allocated in the first incremental update of the day and deleted by the DAILY job (see the INCRDB DYNAM option later in this section). This approach reduces the amount of time that the DASD space is dedicated to incremental update, and lets the amount of DASD space consumed increase through the day as you execute each incremental update.

A third option is to store the incremental update database files on tape (see the INCRDB TAPE option). With this approach, the DASD space is required just for the time that each incremental update or DAILY job step is executing. Note that while this alternative reduces the "permanent" DASD space requirement, the total amount of DASD space required while the incremental update or DAILY jobs are executing is unchanged. In addition, the TAPE option adds processing to copy the incremental update files to tape, and to reload the files from tape to disk.

Note: The incremental update checkpoint file is always a permanently allocated disk data set. This is a small data set and should not be an issue.

- o Operational Complexity

Incremental update expands your measurement data management and job scheduling issues. You must ensure that each incremental update and the DAILY job processes your measurement data chronologically; that is, each job must see data that is newer than the data processed by the prior job. By incrementally updating the database, you have more opportunities to miss a log file, or to process a log out of order.

- o Interval End Effects

Each incremental update processes a subset of the day's measurement data, taking advantage of early availability of some of the day's data, for example, when a measurement log fills and switches to a new volume. This can cause a problem if the measurement log split occurs while the data source is logging records for the end of a measurement interval, thus splitting the data for a single measurement interval across two log files. When an incremental update processes the first log file, the checkpoint high end timestamp is set to indicate that this split measurement interval has been processed. Then, when the rest of the measurement interval's data is encountered in a later update, it can be dropped as duplicate data (because data for this measurement interval end timestamp has already been processed).

Appropriate scheduling of log dumps and incremental updates can avoid this problem. For example, if you plan to run incremental updates at 7:00 AM and 6:00 PM, you could force a log dump in the middle of the measurement interval just prior to the scheduled incremental update executions. This is an extension of the procedure you may already be using for end-of-day measurement log processing. The objective is to ensure that all records for each monitor interval are processed in the same incremental update.

- o Dynamic Allocation

When you activate incremental update and specify TAPE or DYNAM for the INCRDB parameter, dynamic allocation is employed for the incremental update database files. If your site restricts dynamic allocation of large, cataloged data sets, you must use the INCRDETAIL and INCRDAYS parameters to direct incremental update data set allocation to a generic unit or storage class where dynamic allocation is allowed.

o Data Set Names

The incremental update database files are allocated and cataloged according to standard CA MICS unit database data set name conventions. The DDNAME and default data set names are (where ccc is the product ID):

- o Incremental update checkpoint file,  
//IUCKPT DD DSN=prefix.MICS.ccc.IUCKPT,.....
- o Incremental update DETAIL  
//IUDETAIL DD DSN=prefix.MICS.ccc.IUDETAIL,.....
- o Incremental update DAYS  
//IUDAYS DD DSN=prefix.MICS.ccc.IUDAYS,....

Since these data sets conform to the same data set name conventions as your existing CA MICS data sets, there should be few, if any, data-set-name-related allocation issues. However, it is possible to override the data set names if required. Contact Technical Support at <http://ca.com/support> for assistance if you must change data set names.

INCRDB  
-----

This statement is optional. The default is this:

INCRDB PERM

Note: INCRDB is ignored when you specify INCRUPDATE NO.

Specify this statement or take the default, to keep the incremental update database DETAIL and DAYS files on permanently allocated DASD data sets:

INCRDB PERM

Execute the prefix.MICS.CNTL(cccIUALC) job to allocate the incremental update database files.

```
*****  
*                                                                 *  
* Note: The incremental update checkpoint file is always *  
*       a permanently allocated DASD data set.           *  
*                                                                 *  
*****
```

\*\*\*\*\*

Specify this to offload the incremental update DETAIL and DAYS files to tape between incremental update executions:

INCRDB TAPE #gdgs UNIT=name

With the TAPE option, the incremental update DETAIL and DAYS DASD data sets are dynamically allocated at the beginning of the incremental update job or DAILY job step, and then are deleted after the job step completes.

- o The first incremental update job of the day allocates and initializes the incremental update database files. At the end of the job, the DETAIL and DAYS files are copied to a new (+1) generation of the incremental update tape data sets. Then the DASD files are deleted.
- o Subsequent incremental update jobs restore the DASD incremental update database files from the current, (0) generation, incremental update tape data sets before processing the input measurement data. At the end of the job, the DETAIL and DAYS files are copied to a new (+1) generation of the incremental update tape data sets. Then the DASD files are deleted.
- o The DAILY job step also restores the DASD incremental update database files from the (0) generation tape files before processing the input data, but does NOT copy the incremental update database files to tape. Thus, the DAILY job actually creates a new, null (+1) generation.
- o Use the #gdgs parameter to specify the maximum number of incremental update tape generations. The minimum is 2 and the maximum is 99, with a default of 5. Set the number of generations equal to or greater than the number of incremental updates, including the DAILY job you plan to execute each day. This facilitates restart and recovery if you encounter problems requiring you to reprocess portions of the daily measurement data.
- o Use the optional UNIT=name parameter to specify a tape unit name for the incremental update database output tapes. The default is to use the same tape unit as the input tapes.
- o A special index must be created in your system catalog for each of the incremental update tape data set generation

data groups. The prefix.MICS.CNTL(cccIUGDG) job will generate the statements to create the incremental update GDG index definitions.

- Before each index is built, it is deleted. These DLTX (or DELETE) statements causes an error message if no entry exists. This is done so that you can change the number of entries without having to delete each of the index entries.
- DLTX and BLDG (or DELETE and DEFINE) fail if there is a cataloged data set with the same index. IDCAMS (or IEHPROGM) issues a message and gives a return code of 8. This issue is not a problem for non-GDG entries or if the GDG already has the desired number of entries.
- If you want to change the number of entries kept in a GDG with cataloged data sets, do the following:
  1. Uncatalog any existing entries in the GDG.
  2. Delete the index with a DLTX (or DELETE).
  3. Create the index with a BLDG (or DEFINE).
  4. Catalog any entries that are uncataloged in step 1.
- o The incremental update tape data set names are as follows, where ccc is the product ID:
  - Incremental update tape DETAIL file  
tapeprefix.MICS.ccc.IUXTAPE.GnnnnV00
  - Incremental update tape DAYS file  
tapeprefix.MICS.ccc.IUDTAPE.GnnnnV00

```
*****  
*                                                                 *  
* Note: The INCRDETAIL and INCRDAYS parameters are               *  
*         required when you specify INCRDB TAPE.                 *  
*                                                                 *  
*****
```

Specify this parameter to allocate dynamically the incremental update DETAIL and DAYS DASD data sets in the first incremental update of the day, and then delete these data sets at the end of the DAILY job step:

INCRDB DYNAM

- o With this option, no space is used for the incremental update database files during the time between the end of the DAILY job step and the beginning of the next day's first incremental update.
- o With this approach, you can set the data set allocation parameters so that the incremental update DETAIL and DAYS data sets start out with a minimum allocation and then grow through secondary allocations as more space is required for subsequent incremental updates. For example, enough space for one incremental update.

```
*****
*                                                                 *
* Note: The INCRDETAIL and INCRDAYS parameters are             *
*        required when you specify INCRDB DYNAM.                *
*                                                                 *
*****
```

#### INCRDETAIL

-----

This statement is required if you specify either of these:

INCRDB TAPE

INCRDB DYNAM

Otherwise, this statement is optional. There is no default.

Specify this to define data set allocation parameters for the incremental update DETAIL data set (IUDETAIL):

```
INCRDETAIL    data_set_allocation_parameters
```

Note: INCRDETAIL is ignored when you specify INCRUPDATE NO.

The incremental update DETAIL data set (IUDETAIL) contains the current incremental update detail-level database files, and the DETAIL "to-date" data for the current daily update cycle. You should allocate DASD space equivalent to two cycles of this product's DETAIL timespan data.

If you specified INCRDB PERM (the default), your INCRDETAIL parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update database and checkpoint files.
- o Omit the INCRDETAIL parameter if you prefer to specify data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

If you specified INCRDB TAPE or INCRDB DYNAM, your INCRDETAIL parameter specifications are used in incremental update DETAIL data set dynamic allocation during incremental update or DAILY job step execution.

- o The INCRDETAIL parameter is required for the TAPE or DYNAM option.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRDETAIL keyword on the continuation line.
- o INCRDETAIL accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

- o You can override the INCRDETAIL data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Example 1:

```
INCRDETAIL STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
 pp is the primary allocation  
 ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRDETAIL STORCLAS=MICSTEMP UNIT=SYSDA
INCRDETAIL SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
 The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
 The name can have up to eight characters.

SPACE - specifies how much disk space to provide for  
 a new data set being allocated.

INCRDAYS  
 -----

This statement is required if you specify either of these:

INCRDB TAPE

INCRDB DYNAM

Otherwise, this statement is optional. There is no default.

Specify this to define data set allocation parameters for the incremental update DAYS data set (IUDAYS):

```
INCRDAYS data_set_allocation_parameters
```

Note: INCRDAYS is ignored when you specify INCRUPDATE NO.

The incremental update DAYS data set (IUDAYS) contains the current incremental update days-level database files, and the

DAYS "to-date" data for the current daily update cycle. You should allocate DASD space equivalent to two cycles of this product's DAYS timespan data.

If you specified INCRDB PERM (the default), your INCRDAYS parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update database and checkpoint files.
- o Omit the INCRDAYS parameter if you prefer to specify data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

If you specified INCRDB TAPE or INCRDB DYNAM, your INCRDAYS parameter specifications are used in incremental update DAYS data set dynamic allocation during incremental update or DAILY job step execution.

- o The INCRDAYS parameter is required for the TAPE or DYNAM option.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRDAYS keyword on the continuation line.
- o INCRDAYS accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

- o You can override the INCRDAYS data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, Section 2.3.6.

Example 1:

```
INCRDAYS    STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.

The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
pp is the primary allocation  
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRDAYS    STORCLAS=MICSTEMP UNIT=SYSDA
INCRDAYS    SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set. The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

```
INCRCKPT
-----
```

This statement is optional. Specify this to override default data set allocation parameters for the incremental update checkpoint data set:

```
INCRCKPT    data_set_allocation_parameters
```

Note: INCRCKPT is ignored when you specify INCRUPDATE NO.

The incremental update checkpoint data set tracks incremental update job status and the data that has been processed during the current daily update cycle. The incremental update checkpoint is used to detect and block the input of duplicate data during incremental update processing. This data set

will be exactly the same size as prefix.MICS.CHECKPT.DATA (the unit checkpoint data set), usually 20K to 200K depending on the prefix.MICS.PARMS(SITE) CKPTCNT parameter (100-9999).

Your INCRCKPT parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update checkpoint file. If you specified INCRDB PERM, then the cccIUALC job will also allocate the incremental update DETAIL and DAYS database files.
- o By default the incremental update checkpoint data set is allocated as SPACE=(TRK,(5,2)) using the value you specified for the prefix.MICS.PARMS(JCLDEF) DASDUNIT parameter.
- o Omit the INCRCKPT parameter if you prefer to override data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRCKPT keyword on the continuation line.

INCRCKPT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
INCRCKPT    STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen

pp is the primary allocation

ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRCKPT   STORCLAS=MICSTEMP UNIT=SYSDA
INCRCKPT   SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for  
a new data set being allocated.

INCRSPLIT

-----

This statement is optional and defaults to this:

INCRSPLIT IGNORE

Specify the following if you want the incremental update job for this product to get input measurement data from the output of the SPLITSMF job. The optional data\_set\_allocation\_parameters are used by the SPLITSMF job when creating the measurement data file for this product.

INCRSPLIT USE data\_set\_allocation\_parameters

Note: INCRSPLIT is ignored when you specify INCRUPDATE NO.

This option would be used when multiple products in a single unit database are enabled to incremental update. The SPLITSMF job performs the same function for incremental update jobs as the DAILY job DAYSMF step performs for the DAYnnn database update steps.

- o The SPLITSMF job dynamically allocates, catalogs, and populates prefix.MICS.ccc.IUSPLTDS data sets for each product in the unit database for which you specified both the INCRUPDATE YES and INCRSPLIT USE parameters. These data sets are then deleted after processing by the appropriate INCRccc job.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRSPLIT keyword on each continuation line.
- o INCRSPLIT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Specify the following or accept the default if you want the incremental update jobs for this product to get their input measurement data from the data sets specified in the INPUTccc (or INPUTSMF) member of prefix.MICS.PARMS:

INCRSPLIT IGNORE

When you specify INCRSPLIT IGNORE, this product will NOT participate in SPLITSMF job processing.

Example 1:

```
INCRSPLIT USE STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen  
pp is the primary allocation  
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRSPLIT USE  STORCLAS=MICSTEMP UNIT=SYSDA
INCRSPLIT      SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.  
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.  
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for  
a new data set being allocated.

DYNAMWAIT

-----

This statement is optional. Specify the following:

DYNAMWAIT minutes

to override the default amount of time, in minutes, the DAILY and/or INCRccc job will wait for an unavailable data set.

Note: This optional parameter is not normally specified.  
The system default is adequate for most data centers.

Internal Step Restart and Incremental Update facilities use z/OS dynamic allocation services to create new data sets and to access existing data sets. Data set naming conventions and internal program structure are designed to minimize data set contention. However, if data set allocation does fail because another batch job or online user is already using a data set, DAILY and/or INCRccc processing will wait 15 seconds and then try the allocation again. By default, the allocation will be attempted every 15 seconds for up to 15 minutes. After 15 minutes, the DAILY or INCRccc job will abort.

If data set contention in your data center does cause frequent DAILY or INCRccc job failures, and you are unable to resolve the contention through scheduling changes, you may want to use the DYNAMWAIT parameter to increase the maximum number of minutes the DAILY and/or INCRccc jobs will wait for the data set to become available.

On the other hand, if your data center standards require that the DAILY and/or INCRccc jobs fail immediately if required data sets are unavailable, specify the following:

DYNAMWAIT 0

Note: You can override the DYNAMWAIT parameter at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

```
+-----+
| INSTALLATION PREPARATION WORKSHEET: HSM Options Definition |
| | |
| PARMs Library Member is HSMOPS |
| Reference: Section 7.3.2 |
+-----+
| HSM PROCESSING OPTIONS: |
| | |
| ACCOUNTING _____ (DETAIL or DAYS) |
| | |
| OPTIONS _____ (sysid) |
| | |
| | _____ (default duration) |
| | |
| WORK _____ (optional: n data_set_allocation_parameters) |
| | |
| RESTART _____ (optional: YES/NO) |
| | |
| INCRUPDATE _____ (optional: YES or NO) |
| INCRDB _____ (optional: PERM, TAPE, or DYNAM) |
| INCRDETAIL _____ (optional: data_set_allocation_parameters) |
+-----+
```



### 7.3.3 INPUTRDR, INPUTVCA, and INPUTHSM PARMS Members

The prefix.MICS.PARMS members INPUTRDR and INPUTccc are used to specify the input data for the Data Integration Application component step.

To determine whether INPUTccc or INPUTRDR or both are used, review the DAYSMF, SMFRECORDING, and SMFDRCTR specifications in prefix.MICS.PARMS(JCLDEF). Consider the following table and comments for the row:

ROW	PARAMETER IN JCLDEF	INPUTRDR WHERE USED	INPUTccc WHERE USED
1	DAYSMF OFF	Not Used	Each Comp Step
2	DAYSMF FILE(S) ...	DAYSMF Step	Not Used
3	Only 1 SMF Comp	Comp Step	Not Used
4	SMFDIRECTOR	Not Used	Comp Step
5	DAYSMF EXCLUDE ccc	Not Used	Comp ccc Step
6	SMFRECORDING ccc	It depends	Comp ccc Step

Row 1: When DAYSMF OFF is specified in JCLDEF, then the prefix.MICS.PARMS(INPUTccc) member is used for each component step.

Row 2: When DAYSMF FILES PERMANENT or TEMPORARY is specified in JCLDEF, the prefix.MICS.PARMS(INPUTRDR) member is used in DAYSMF.

Row 3: If there is only one SMF recording component in this unit, DAYSMF will not be generated and the component step with SMF input data will use the prefix.MICS.PARMS(INPUTRDR) member.

Row 5: If the use of DAYSMF EXCLUDE ccc results in only one component remaining as DAYSMF eligible, DAYSMF will automatically be deactivated, and row 1 applies.

Row 6: Some components accept data from both SMF and non-SMF sources, such as CIC and VCA. In this case, the component may optionally input data from the INPUTRDR member, but will always include the INPUTccc member. Refer to the component's guide.

For details on the DAYSMF parameters, see PIOM section 2.3.3.2.1, JCL Option Definitions (JCLDEF).

For details on deactivating DAYSMF, see PIOM section 5.10, Removing the DAYSMF Step from the DAILY Job.

The following sections describe how to specify the input data set JCL definitions.

- 1 - INPUTRDR PARMS Member JCL Definition
- 2 - INPUTVCA PARMS Member JCL Definition
- 3 - INPUTHSM PARMS Member JCL Definition

### 7.3.3.1 INPUTRDR PARMS Member JCL Definition

The INPUTRDR member of prefix.MICS.PARMS defines the DD statements that specify the SMF input data for all products in the units that use SMF data. The INPUTSMF DD statements are used by the DAYSMF step, which reads and splits the SMF data into separate files, one for each CA MICS product.

The INPUTRDR member defines the input SMF data sets when DAYSMF FILES PERMANENT or TEMPORARY is specified in the prefix.MICS.PARMS(JCLDEF) member.

An INPUTSMF DD statement is required for each SMF data source. A worksheet for preparing the INPUTRDR member is provided below. If you change the contents of the INPUTRDR member, you must regenerate the CA MICS DAILY job using either JCLGENU or JCLGEND in prefix.MICS.CNTL.

For example, if you want to read your SMF data from a data set named SMF.DAILY.DATA, you would change the INPUTRDR member to contain the name of the SMF data set as follows:

```
//INPUTSMF DD DISP=SHR,DSN=SMF.DAILY.DATA
```

If there is only one SMF recording component in this unit, DAYSMF will not be generated and the component step with SMF input data will use the prefix.MICS.PARMS(INPUTRDR) member directly.

If DAYSMF has been deactivated by the DAYSMF OFF statement specified in the prefix.MICS.PARMS(JCLDEF) member, then each component step will read its input from their corresponding prefix.MICS.PARMS(INPUTccc) member.

```
+-----+
| INSTALLATION PREPARATION WORKSHEET: SMF Input DD Statements |
| PARS Library Member is INPUTRDR |
+-----+
|
| This definition is required to specify the DD statement for SMF data |
| which will be read by the DAILY CA MICS job. |
|
| //@ |
| //@ WARNING: ALWAYS MAKE CHANGES IN PARS(INPUTRDR) AND NOT |
| //@ &CNTL(DAILY). |
| //@ CHANGES MADE TO &CNTL(DAILY) |
| //@ WILL BE GONE WHEN DAILY REGENERATED BY JCLGEN. |
| //@ |
| //INPUTSMF DD DISP=SHR,DCB=BUFNO=3,DSN=_____ |
| // DD DISP=SHR,DCB=BUFNO=3,DSN=_____ |
|
+-----+
```

Figure 7-8a. INPUTRDR JCL Definition Worksheet

### 7.3.3.2 INPUTVCA PARMS Member JCL Definition

The INPUTVCA member of prefix.MICS.PARMS contains the DD statements to specify the input data for the Space Analyzer. The //INPUTSMF DD and, optionally, the //SMFDRCTR DD statements are used by the DAY090 step of the DAILY job.

The INPUTVCA member contains the input SMF data sets when ANY of the following JCLDEF options is used:

- o DAYSMF OFF is specified
- o SMFDRCTR/SMFDIRECTOR VCA is specified
- o DAYSMF EXCLUDE VCA is specified

A //INPUTSMF DD statement is required for each SMF data source.

A //SMFDRCTR DD statement is only required when CA SMF Director split indices are used.

A worksheet for preparing the INPUTVCA member is provided below in Figure 7-8b.

If you change the contents of the INPUTVCA member, you must regenerate the DAILY job using either JCLGENU or JCLGEN in prefix.MICS.CNTL.

The INPUTVCA member can contain JCL definitions for non-SMF input such as data from independent sequential data sets or it can contain JCL definitions for SMF input if you do not specify 'SMFRECORDING VCA' in prefix.MICS.PARMS(JCLDEF). If 'SMFRECORDING VCA' is specified in JCLDEF, then it must be defined in the input JCL in the INPUTRDR member in prefix.MICS.PARMS.

NOTE: If your site has CA SMF Director installed, you can use it to extract specific SMF record types and subtypes at SMF dump time. For more details on this product feature see section 6.1, CA MICS and CA SMF Director Interface of this guide.

```
+-----+
| INSTALLATION PREPARATION WORKSHEET: INPUTVCA JCL Definitions
| PARS Library Member is INPUTVCA
+-----+
|
| This definition is required to specify the DD statement for data
| that will be read by the DAILY CA MICS job.
|
| //@
| //@ WARNING: ALWAYS MAKE CHANGES IN PARS(INPUTVCA) AND NOT
| //@ &CNTL(DAILY).
| //@ CHANGES MADE TO &CNTL(DAILY)
| //@ WILL BE GONE WHEN DAILY REGENERATED BY JCLGEN.
| //@
| //INPUTSMF DD DISP=SHR,DCB=BUFNO=5,DSN=_____
| // DD DISP=SHR,DCB=BUFNO=5,DSN=_____
| //@
| //SMFDRCTR DD DISP=SHR,DCB=BUFNO=5,DSN=_____
| // DD DISP=SHR,DCB=BUFNO=5,DSN=_____
+-----+
```

Figure 7-8b. INPUTVCA JCL Definition Worksheet

### 7.3.3.3 INPUTHSM PARMS Member JCL Definition

The INPUTHSM member of prefix.MICS.PARMS contains the DD statements to specify the input data for the Space Analyzer. The //INPUTSMF DD and, optionally, the //SMFDRCTR DD statements are used by the DAY095 step of the DAILY job.

The INPUTHSM member contains the input SMF data sets when ANY of the following JCLDEF options is used:

- o DAYSMF OFF is specified
- o SMFDRCTR/SMFDIRECTOR HSM is specified
- o DAYSMF EXCLUDE HSM is specified

A //INPUTSMF DD statement is required for each SMF data source.

A //SMFDRCTR DD statement is only required when CA SMF Director split indices are used.

A worksheet for preparing the INPUTHSM member is provided below in Figure 7-9.

If you change the contents of the INPUTHSM member, you must regenerate the DAILY job using either JCLGENU or JCLGEND in prefix.MICS.CNTL.

The INPUTHSM member can contain JCL definitions for non-SMF input such as data from independent sequential data sets or it can contain JCL definitions for SMF input if you do not specify 'SMFRECORDING HSM' in prefix.MICS.PARMS(JCLDEF). If 'SMFRECORDING HSM' is specified in JCLDEF, then it must be defined in the input JCL in the INPUTRDR member in prefix.MICS.PARMS.

NOTE: If your site has CA SMF Director installed, you can use it to extract specific SMF record types and subtypes at SMF dump time. For more details on this product feature see section 6.1, CA MICS and CA SMF Director Interface of this guide.

```
+-----+
| INSTALLATION PREPARATION WORKSHEET:  INPUTHSM JCL Definitions
| PARS Library Member is INPUTHSM
+-----+
|
| This definition is required to specify the DD statement for data
| that will be read by the DAILY CA MICS job.
|
| //@
| //@ WARNING: ALWAYS MAKE CHANGES IN PARS(INPUTHSM) AND NOT
| //@           &CNTL(DAILY).
| //@           CHANGES MADE TO &CNTL(DAILY)
| //@           WILL BE GONE WHEN DAILY REGENERATED BY JCLGEN.
| //@
| //INPUTSMF DD DISP=SHR,DCB=BUFNO=5,DSN=_____
| //          DD DISP=SHR,DCB=BUFNO=5,DSN=_____
| //@
| //SMFDRCTR DD DISP=SHR,DCB=BUFNO=5,DSN=_____
| //          DD DISP=SHR,DCB=BUFNO=5,DSN=_____
|
+-----+
```

Figure 7-9. INPUTHSM JCL Definition Worksheet

### 7.3.4 Database Space Modeling (DBMODEL)

This section describes the information required to define the Analyzer to the Database Space Modeling Facility. Specifically, the user must provide values for the cycle (data retention) definitions. The special considerations for defining the FILE input statements to the Database Space Modeling facility are discussed.

Section 2.3.4 of the PIOM, Database Space Modeling Facility, explains how these values are used in estimating the DASD requirements of the database, and how the user can input these values to the modeling facility.

This section covers the following topics:

- 1 - Data Retention Specifications (FILE Statements)
- 2 - DBMODEL Input Statements

### 7.3.4.1 Data Retention Specifications (FILE Statements)

Data retention specifications tell the CA MICS database how many cycles of each file in each supported timespan are to be saved both in online and archive modes. Figure 7-10 is the worksheet for collecting this information.

The worksheet is organized by information area. Each file in the area is listed by name. For each file, a line is formatted to allow six definitions in the online database and two in the archive database:

- o The online database files quantify the number of cycles of data that is maintained in the DETAIL, DAYS, WEEKS, MONTHS, and YEARS timespans and the TABLES data area.
- o The two definitions for the archive database files quantify the number of cycles of data to be retained, up to the cutoff limit defined. The archive definitions have no impact on the size of the database and can be specified whethert the weekly and/or monthly archive history files have actually been activated (see Section 2.3.3, CA MICS JCL Planning and Parameters, of the PIOM).

The worksheet formats provide an underscored area for the user's definition, followed by the recommended value, shown within parentheses. If the underscored area contains a value of 00, the file is not supported for the indicated timespan. To add support, perform database tailoring as described in Section 6.2, Tailoring the Database, of the System Modification Guide (SMG).

When specifying a retention limit, remember that the number may never be zero if the file has been defined to be active in the timespan.

#### EXAMPLE

The FILE statements listed below illustrate how to enter the information:

```
FILE VCA VCADAA 10 33 00 06 00 02 000 024
FILE VCA VCA_VS 10 33 00 06 00 02 000 024
FILE VCA VCAVOA 10 33 00 06 00 02 000 024
FILE VCA VCADCS 00 33 00 00 00 00 000 000
FILE VCA HFSDIR 10 33 00 06 00 00 000 024

FILE HSM HSM MIG 10 33 00 06 00 02 000 024
FILE HSM HSMVOL 10 33 00 06 00 02 000 024
```

```
FILE HSM HSMTAP 60 33 00 06 00 02 000 024
FILE HSB HSBAC 10 33 00 06 00 02 000 024
```

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: Database Data Retention Definitions |
| PARS Library Member is DBMODEL |
| Reference: Section 7.3.4.1 |
+-----+
| File | Online Database Retention | Archive Cutoff |
| Name | DETAIL | DAYS | WEEKS | MONTHS | YEARS | TABLES | WEEKS | MONTHS |
+-----+
|HFSDIR| |(10) |(33) |00(00) | |(06) | 0(2) | 00(00) |000 (000) | |(24) |
|VCADAA| |(10) |(33) |00(00) | |(06) | 0(2) | 00(00) |000 (000) | |(24) |
|VCA_VS| |(10) |(33) |00(00) | |(06) | 0(2) | 00(00) |000 (000) | |(24) |
|VCAVOA| |(10) |(33) |00(00) | |(06) | 0(2) | 00(00) |000 (000) | |(24) |
|VCADCS| |00(00) |(33) |00(00) | |(00) | 0(0) | 00(00) |000 (000) | |000 (000) |
+-----+
|HSM MIG| |(10) |(33) |00(00) | |(06) | 0(2) | 00(00) |000 (000) | |(24) |
|HSMVOL| |(10) |(33) |00(00) | |(06) | 0(2) | 00(00) |000 (000) | |(24) |
|HSMTAP| |(60) |(33) |00(00) | |(06) | 0(2) | 00(00) |000 (000) | |(24) |
|HSBAC| |(10) |(33) |00(00) | |(06) | 0(2) | 00(00) |000 (000) | |000 (000) |
+-----+

```

Figure 7-10. Data Retention Specifications Worksheet

### 7.3.4.2 DBMODEL Input Statements

The DBMODEL member of prefix.MICS.PARMS provides the input to the Database Space Modeling Facility.

Update the DBMODEL member using the information collected on the worksheet in Figure 7-10. To actually perform the space modeling, submit the jobs as described in Section 2.3.4.2 of the CA MICS Planning, Installation, Operation, and Maintenance Guide (PIOM).

# Chapter 8: INSTALLATION

---

After specifying the parameters documented in Chapter 7 of this guide, you can install the CA MICS Space Analyzer using the checklists in Section 3.8 of the PIOM.

Please note that when the PIOM refers to the product in terms of its "component identifier" or "ccc", this should be read to mean VCA and (optionally) HSM. That is, when installing the CA MICS Space Analyzer, you will have to repeat the checklist you did for VCA when you need to generate HSM. A different series of parameters and code is generated to support HSM as an optional "component" within the product.

Conversely, it is not expected that VCA would be construed as optional within the context of the product. VCA is assumed while if you are using a non-IBM data management product you may not require the HSM component.



# Chapter 9: PROCESSING

---

Creating and maintaining a large database system such as CA MICS requires a closely monitored approach to the periodic updating processes. The CA MICS database, just like any other information database (such as inventory control or personnel), requires a systematic, monitored update process.

This chapter describes the daily processing flow of the DASD Space Analyzer, the exit points available during the daily update, data screening reports written to the CA MICS log, and considerations for running the daily DAY090 and DAY095 update steps.

This section contains the following topics:

[9.1 Processing Overview](#) (see page 300)

[9.2 Daily Processing Flow for VCA \(DAY090\)](#) (see page 306)

[9.3 Daily Processing Flow for HSM and HSB \(DAY095\)](#) (see page 319)

## 9.1 Processing Overview

The CA MICS Space Analyzer processes data captured by the CA MICS Space Collector (VCC). VCC obtains the data by sequentially reading the VTOC (volume table of contents) from DASD volumes and by sequentially reading the system catalog structures (both the BCS and the WVDS). VCC can also obtain the available data from the DFHSM Migration and Backup Control Data Sets (MCDS and BCDS, respectively) by using the ARCUTIL collection facility.

Because of the probability for a large amount of data from the aforementioned sources, support for the Space Analyzer has been implemented by means of two separate steps of the CA MICS DAILY job and three separate information areas. The DAY090 step processes information into the VCA Information Area, and the DAY095 step processes information into the HSM and HSB Information Areas. This allows you to overlap processing by defining different unit databases for the VCA, HSM, and HSB information areas.

The major concepts and processes involved in collecting the input SMF data via VCC and updating the CA MICS database are described in the following sections:

- 1 - Relationship to VCC SMF Record Production
- 2 - Updating the VCA Information Area
- 3 - Updating the HSM and HSB Information Areas

### 9.1.1 Relationship to VCC SMF Record Production

VCC provides the raw input data to the Space Analyzer in SMF format. The records can be written to separate data sets for direct input into the daily update steps (DAY090 and DAY095), or they can be written to the standard system SMF data sets. See the CA MICS Space Collector Guide for more information on the creation and format of the records, as well as the options available for tailoring the collection process.

Two different SMF record types are produced by VCC for input to the Space Analyzer. The defaults are SMF record type 198 for VCA and type 197 for HSM (and HSB).

The type 198 record containing information for building the VCA Information Area is segmented in such a way that different types of data can be represented by one record. The different segment types contain overall configuration information, non-VSAM data set information, VSAM data set information, and VTOC/volume information, which is then used to construct the various CA MICS files.

The type 197 record is divided into different subtypes, such that each subtype addresses a different DFHSM resource. There are five different subtype records, identified as:

- subtype M (a migrated data set)
- subtype B (a backup version)
- subtype C (a DASD volume statistics record)
- subtype T (a record of DFHSM-managed tape volumes)
- subtype Z (a VCC-generated inventory trailer record)

As with the type 198 record, various CA MICS files are constructed from the information contained in the different subtypes.

## 9.1.2 Updating the VCA Information Area

The Space Analyzer makes use of the standard database update and summarization facilities of CA MICS to maintain its data in the online and offline databases. Data is supplied to CA MICS in the DAY090 step of the DAILY database update job.

The daily update step:

- o Reads and formats the raw data
- o Maintains data integrity by eliminating duplicate data
- o Adds new cycles to the DETAIL timespan
- o Summarizes activity by hour for DAYS timespan
- o Adds new cycles to the DAYS timespan
- o Updates the month-to-date cycles

Figure 9-1 shows an overview of the daily update process. A more detailed description of DAY090 processing is presented in Section 9.2.

Weekly processing is performed by the WEEK090 and WEEK300 steps. The WEEK090 step splits the week-to-date files into a new week-to-date and a new weekly '01' cycle. The WEEK300 step creates the weekly offline database files.

Monthly processing is performed by the MONTH090 and MONTH300 steps. The MONTH090 step splits the month-to-date files into a new month-to-date and a new monthly '01' cycle, and also updates the year-to-date cycles. The MONTH300 step creates the monthly offline database files.

Yearly processing is performed by the YEAR090 step. The YEAR090 step splits the year-to-date files into a new year-to-date and a new yearly '01' cycle.

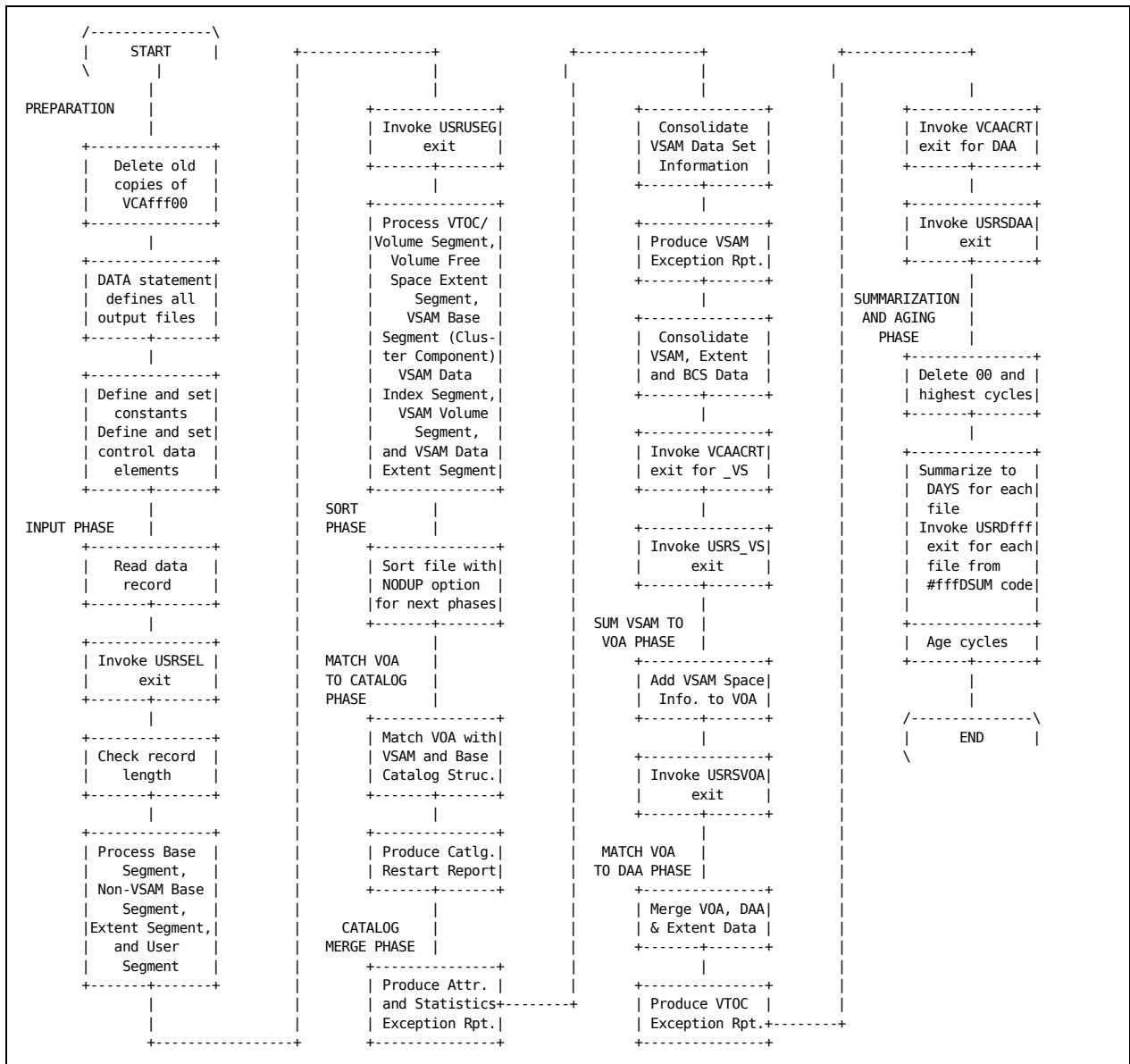


Figure 9-1. VCA DAY090 Processing Flow Diagram

### 9.1.3 Updating the HSM and HSB Information Areas

The Space Analyzer makes use of the standard database update and summarization facilities of CA MICS to maintain its data in the online and offline databases. SMF format data provided by VCC is supplied to CA MICS in the DAY095 step of the DAILY database update job.

The daily update step:

- o Reads and formats the raw data
- o Maintains data integrity by eliminating duplicate data
- o Adds new cycles to the DETAIL timespan
- o Summarizes activity by hour for DAYS timespan
- o Adds new cycles to the DAYS timespan
- o Updates the week-to-date cycles
- o Updates the month-to-date cycles

Figure 9-2 shows an overview of the daily update process. A more detailed description of DAY095 processing is presented in Section 9.3.

Weekly processing is performed by the WEEK095 and WEEK300 steps. The WEEK095 step splits the week-to-date files into a new week-to-date and a new weekly '01' cycle. The WEEK300 step creates the weekly offline database files.

Monthly processing is performed by the MONTH095 and MONTH300 steps. The MONTH095 step splits the month-to-date files into a new month-to-date and a new monthly '01' cycle, and also updates the year-to-date cycles. The MONTH300 step creates the monthly offline database files.

Yearly processing is performed by the YEAR095 step. The YEAR095 step splits the year-to-date files into a new year-to-date and a new yearly '01' cycle.

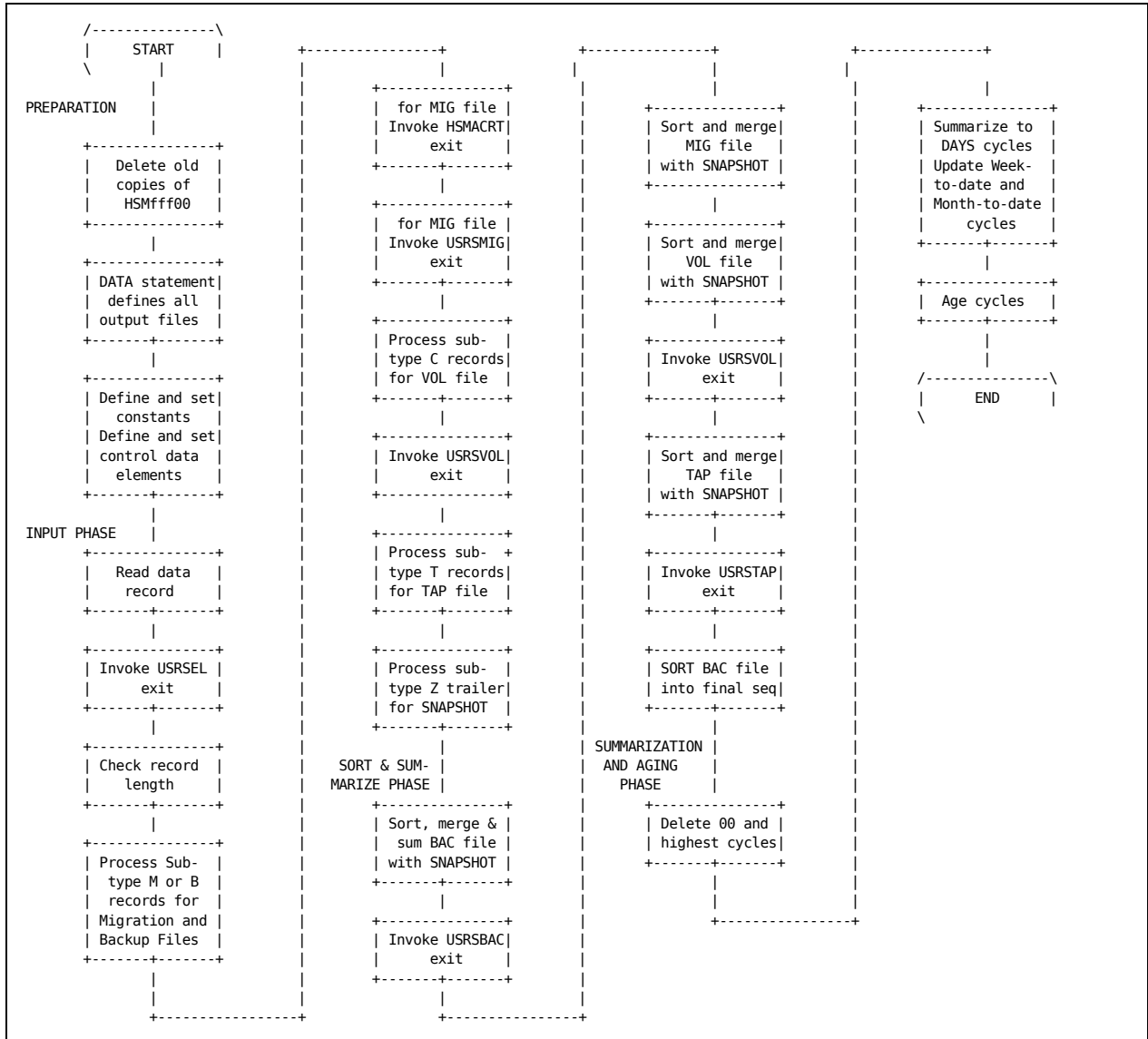


Figure 9-2. HSM/HSB DAY095 Processing Flow Diagram

## 9.2 Daily Processing Flow for VCA (DAY090)

This section describes the daily processing flow for the VCA and HFS Information Area and its relation to the locations of the VCA user exit routines. For specific questions not discussed in this document, see the source code referenced by the program module.

The CA MICS VCA update process (the DAY090 step of the DAILY update job) is divided into seven distinct processing phases. During several of the phases (phases 3, 4, 5 and 6), data screening performed by the input routine may produce reports to the MICSLOG describing unusual or exceptional conditions encountered. The following sections describe the MICSLOG reports and the general logic flow in each of the phases, as well as the location of standard user routines or user exits.

- 1 - MICSLOG Reporting
- 2 - Phase 1 - Input Raw VCC and/or HFS SMF Data
- 3 - Phase 2 - Sorts
- 4 - Phase 3 - Match VOA to Catalog
- 5 - Phase 4 - Merge Catalog Information
- 6 - Phase 5 - Summarize VSAM Information into VOA
- 7 - Phase 6 - Match VOA to DAA
- 8 - HFS - Sort, match, and merge HFS files
- 9 - Phase 7 - Summarization and Aging

## 9.2.1 MICCSLOG Reporting

The MICCSLOG Sysout data set is used to display information on the input sources and progress of the Daily update step DAY090. An example of the MICCSLOG data set produced by a typical Daily update job is shown in Figure 9-3. The source of the input for this run was SYSID PRD1, and no records were excluded from the source file due to invalid record conditions. The progress of each processing phase is indicated by the message time stamps, and the checkpoint file is updated to reflect the data processed during the run.

Because a number of error conditions can be detected during the processing of the raw input data, several reports can be produced by different processing phases. Figure 9-6 contains an extract of a report produced during Phase 6 of the daily update step in which the DAA and VOA files are matched. The following reports can be produced on the MICCSLOG Sysout data set during the DAY090 step of the Daily Update Job:

- o Catalog Restart Report - Produced during Phase 3 of the DAY090 step (Match VOA to Catalog Phase), this report shows volumes that were not successfully processed by VCC, and indicates the cataloged VSAM data sets residing on such volumes that will not be added to the \_VS file.
- o Attributes and Statistics Exceptions Report - When one or more primary volumes for VSAM data sets are not processed by VCC, this report is generated during Phase 4 of the DAY090 step (Catalog Merge Phase). Since the attributes and statistics elements for the affected VSAM files are only kept in the primary volume VSAM Volume Data Set (VVDS), many elements in the \_VS file will be zero for these observations. The report indicates the affected volume(s) and data set(s).
- o VSAM Exception Report - Produced during Phase 4 of the DAY090 step (Catalog Merge Phase), this report lists those VSAM objects that do not have VCC records from both the Base Catalog Structure (BCS) and a VVDS, or are duplicated in the BCS, VVDS, and/or old-style VSAM catalog.
- o VSAM Volume Exception Report - When volumes containing VSAM data sets are not processed by VCC, the collected information on the individual VSAM data sets residing on the volumes is saved in the \_VS file, but is not summarized to the VOA file. This report is generated

during Phase 5 of the DAY090 step (Sum VSAM to VOA Phase). It lists the VSAM data sets that were not summarized into the VOA file.

- o DFSMS Exception Report / VTOC Restart Report - This combined report identifies possible error conditions resulting from the VCC collection process, or from inconsistencies with the DFSMS volume or data set flags encountered during phase 6 of the DAY090 step (Match VOA to DAA Phase). Examples of types of exceptions processed include the lack of a DFSMS WVDS record, a missing DFSMS VTOC indicator, or an inconsistency, such a data set that appears in the WVDS, but not in the VTOC.
- o USS File Systems Reject Report - Produced during Phase 3 of the DAY090 step, matches VCAFIL work file to VCAHCK file. If a match is not found, then the observation is rejected and file system DSNAME is listed in the MICSLLOG. This report shows all file systems that were not successfully processed by USS File Systems Collector.
- o USS Directory Entry File Reject Report - Produced during Phase 3 of the DAY090 step, matches HFSDIR and VCAHCK file. If a match is not found then the observation is rejected and SYSID, RSRCEID, HFSRUN, VOLSER, and TASKID information for the rejected entry are listed in the MICSLLOG. This report shows all the USS Directory Entries that were not successfully processed by USS File Systems Collector.

```

----- MICSLOG ----- 19:00 SATURDAY, MAY 11, 1991
19.00.43 BAS00119I AT CHECKPOINT FOR STEP "DAY090" IN JOB "DAILY" FOR DATABASE "VCAUNIT".
19.00.43 BAS00122I REQUIRED PREDECESSOR STEP IS:
19.00.43 BAS00122I     PROCESS=DAY
19.00.43 BAS00122I     STEP   =000
19.00.43 BAS00125I CHECKPOINT STEPS: D_STEP=000 W_STEP=900 M_STEP=900 Y_STEP=900 B_STEP=900 R_STEP=900
19.00.43 BAS00125I

19.01.11 BAS00318I ORGSYSID/COMPONENT CHECKPOINT INFORMATION BEFORE PROCESSING
19.01.11 BAS00318I

19.01.12 VCA00100I ----- BEGIN PHASE 1 -----
19.01.12 VCA00011I VCAPGEN PARAMETERS IN EFFECT:
19.01.12 VCA00011I >>>> ACCOUNTING DETAIL
19.01.12 VCA00011I >>>> BCSREQUIRED NO
19.01.12 VCA00011I >>>> EXTENTDETAIL DAA NO
19.01.12 VCA00011I >>>> EXTENTDETAIL _VS NO
19.01.12 VCA00104I FIRST INPUT TIMESTAMP THIS RUN=10MAY91:15:31:52.57 FROM SYSTEM PRD1.
19.01.12 VCA00101W AT LEAST ONE RECORD WITH SYSID NOT DEFINED IN VCAOPS (PRD1).
19.01.12 VCA00101W THE DEFAULT DURATION WILL BE USED.
19.01.47 BAS00326I INPUT FILE(S) CONTAINED 33619 RECORDS FROM 10MAY1991:15:31:52.57 THRU 10MAY1991:15:31:52.57
19.01.47 BAS00326I
19.01.47 BAS00331I DATA SELECTION STATISTICS BY SYSTEM/COMPONENT
19.01.47 BAS00331I
19.01.47 BAS00331I
19.01.47 BAS00331I     SYSTEM COMPONENT     INPUT     DATE RANGE     DELETED     OPTION     KEPT
19.01.47 BAS00331I     PRD1 VCA                33619         0             0           0         33619
19.01.47 BAS00343I
19.01.47 BAS00337I TOTAL RECORDS KEPT = 33619

19.01.49 VCA00200I ----- BEGIN PHASE 2 -----
19.02.41 VCA00300I ----- BEGIN PHASE 3 -----
19.02.55 VCA00400I ----- BEGIN PHASE 4 -----
19.03.52 VCA00500I ----- BEGIN PHASE 5 -----
19.04.03 VCA00600I ----- BEGIN PHASE 6 -----

19.05.18 VCA00999I ----- BEGIN SUMMARIZATION AND AGING -----
19.07.09 BAS00339I CHECKPOINT FILE ORGSYSID/COMPONENT UPDATE FOR COMPONENT VCA
19.07.09 BAS00340I BEFORE UPDATE:
19.07.09 BAS00341I AFTER UPDATE: PRD1 VCA      10MAY91:15:31:52.57 10MAY91:15:31:52.57
19.07.09 BAS00341I
19.07.15 BAS00121I AT FINAL POST FOR STEP "DAY090" IN JOB "DAILY"

```

Figure 9-3. DAY090 MICSLOG Standard Output

## 9.2.2 Phase 1 - Input Raw VCC and HFS SMF Data

The Input Phase reads data records from the file defined by the INPUTSMF DD statement. The user exit USRSEL is executed if defined. The record type is checked and non-VCA records are ignored. The record length is checked to verify that the record is of the proper format. A check is also made to ensure that the record was produced by VCC at the proper level. Several levels of collector data are supported by the DAY090 processing enabling you to service the output from many different sites or collectors with one version of VCA:

- Level 1.0.1 pre-VCA4100
- Level 1.0.2 VCA4100 (initial DFSMS support)
- Level 1.0.3 VCA4200 (support for storage group name)
- Level 1.0.4 VCA4225 (support for catalog status check)

The level numbers are contained in the data records. Generally, new data is added to the end of raw data segments and therefore, will simply not be read up by VCA if not present in a given version level of the record.

During the input phase, the checkpoint file is checked to prevent data which has already been input to CA MICS from being input to CA MICS a second time.

An integral part of this first phase is the creation of a series of work files corresponding to the different segments that might appear in each record produced by VCC. For example, the NON-VSAM Base Extent segment creates a record in either the WORK.VCADAA file or the WORK.EXTDAA file, depending on whether there is a data extent segment indicating a data set with more than 16 extents. Also included in this processing is the handling of any user segment data that might have been extracted from the Format 1 DSCB via the VCCUEXIT (refer to the CA MICS Space Collector Guide). The user exit USRUSEG is invoked, if available, to handle the user segment.

Once all of the input data has been processed, control is passed to the next phase for sorting of the created work files.

### 9.2.3 Phase 2 - Sorts

All of the work files are sorted in order to prepare for the next processing phases. Except for the Base Catalog Structure work file (WORK.VCABCS) the files are sorted by SYSID, ENDTS, VOLSER and DSNAME. The BCS file contains records for each VSAM cluster and data component type (DATA, INDEX or SSET). Consequently, the elements containing these items (DAACNAME and DAAVCOMP respectively), are included in the sort sequence.

In all cases, the NODUP option for PROC SORT is used to purge duplicate data that may have inadvertently been combined in the input job stream.

### 9.2.4 Phase 3 - Match VOA to Catalog

The work files containing VSAM, VSAM extent and Base Catalog data are brought together in this phase along with the Volume Allocation File (VOA).

The primary purpose of this phase is to match a WORK.VCAVOA observation with the intermediate WORK files involved in building VCA\_VS data. If there is not an observation from the WORK.VCAVOA file that matches on VOLSER with these other files, then an anomaly exists, most likely traceable to data collection time.

Refer to Appendix A for the explanation of messages issued in Phase 3 processing.

The intent behind this check and the reporting to MICSL0G is that failure to match up on VOLSER indicates the possibility of data logged to SMF interrupted by a job or system failure but followed later by a successful rerun (or "restart") of the collector. Hence, the DAY090 processor discards these records without a matching VOA observation under the premise that the collection was incomplete and a subsequent new inventory is presented to DAY090 in this same run or even the next day.

A secondary purpose of this phase is to replicate the storage group name for the matching VOLSER to all output observations destined for the VCA\_VS file.

### 9.2.5 Phase 4 - Merge Catalog Information

This phase is actually composed of two distinct parts. In the first part, elements derived from the Access Method Data Statistics Block (AMDSB) for VSAM data sets is replicated across observations for different VSAM components/clusters and or multi-volume data sets. If the AMDSB information from the primary definition is not found for one or more data sets, the Attributes and Statistics Exception Report is generated to the MICSL0G Sysout data set.

Refer to the sample report in Figure 9-4 for more description of this situation. Also, Section 5.2.2.3 describes the effect of this on the VCA\_VS file.

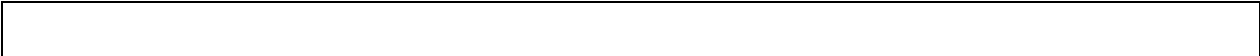
In the second part of this phase, the detail \_VS (VSAM Data Set Allocation) file is created by merging the three work files VCA\_VS, EXT\_VS, and VCABCS. DSNODES 1 through 5 and DSNODEL are created for use by the VCAACRT Accounting routine. The VCA Accounting Routine Exit is invoked once per data set per volume, and the User output exit, USRS\_VS, is also invoked. If EXTENTDETAIL was requested for VSAM data sets having more than 16 extents (See Section 7.3.1), multiple observations are output describing each set of 16 extents. If there is an error in the merge process such that a VSAM object was not found by VCC in the BCS and the VVDS, or is duplicated in the BCS, VVDS and/or old-style VSAM catalog, the VSAM Exception Report is written to the MICSL0G Sysout Data Set, and the observation causing the error condition is dropped from the output file.

Refer to the sample report in Figure 9-5 for more description of this situation. Also, Section 5.2.2.3 describes the effect of this on the VCA\_VS file.

The final processing done in this phase is to re-sort the \_VS file into its final ordering, and write it out to the CA MICS database.

```
----- MICSLOG ----- 10:30 SUNDAY, JUNE 2, 1991
10.34.10 VCA00401W          ATTRIBUTES AND STATISTICS EXCEPTION REPORT
10.34.10 VCA00401W
10.34.10 VCA00401W
10.34.10 VCA00401W          THE PRIMARY VOLUME FOR THE FOLLOWING VSAM DATA SETS WAS NOT PROCESSED BY VCC.  SINCE
10.34.10 VCA00401W          THE ATTRIBUTES AND STATISTICS ELEMENTS ARE ONLY PRESENT IN THE PRIMARY VOLUME VVDS,
10.34.10 VCA00401W          MANY ELEMENTS WILL BE ZERO IN THE VCA_VS FILE.
10.34.10 VCA00401W
10.34.10 VCA00401W TYPE  VOLUME  CLUSTER NAME          DATA SET NAME
10.34.10 VCA00401W -----  -----  -----
10.34.10 VCA00403W INDEX MVS340  BUNDL.V40TEST.VSAM.U9260477  BUNDL.V40TEST.VSAM.KRWFAIF.INDEX
```

Figure 9-4. DAY090 Attributes and Statistics Exception Report



```

----- MICSLOG ----- 10:30 SUNDAY, JUNE 2, 1991
10.34.47 VCA004111 VSAM EXCEPTION REPORT
10.34.47 VCA004111
10.34.47 VCA004111
10.34.47 VCA004111 THE FOLLOWING VOLUMES CONTAIN VSAM OBJECTS THAT DO NOT HAVE VCC RECORDS FROM BOTH
10.34.47 VCA004111 A BASE CATALOG STRUCTURE (BCS) AND A VSAM VOLUME DATA SET (VVDS) OR THAT ARE
10.34.47 VCA004111 DUPLICATED IN AN OLD-STYLE VSAM CATALOG, BCS, AND/OR VVDS.
10.34.47 VCA00412I TYPE (CAT) CLUSTER NAME REASON
10.34.47 VCA00412I DATA SET NAME CATALOG NAME
-----
10.34.47 VCA00413W >>>> VOLUME=MVSC1A - SYSID=S008 31MAY91:21:01:50.70 - DATA SETS HAVE BEEN DELETED FROM VCA_VS
10.34.47 VCA00413W
10.34.47 VCA00415I DSIJMVA.DUMP.DIR NO MATCHING VVDS RECORD
10.34.47 VCA00417I DATA (B) DSIJMVA.DUMP.DIR.DATA CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.47 VCA00415I DSIJMVA.DUMP.DIR NO MATCHING VVDS RECORD
10.34.47 VCA00417I INDEX (B) DSIJMVA.DUMP.DIR.INDEX CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.48 VCA00413W >>>> VOLUME=MVS340 - SYSID=S008 31MAY91:21:01:50.70 - DATA SETS HAVE BEEN DELETED FROM VCA_VS
10.34.48 VCA00413W
10.34.48 VCA00415I BUNDL.V40TEST.VSAM.U9260477 NO MATCHING VVDS RECORD
10.34.48 VCA00417I DATA (B) BUNDL.V40TEST.VSAM.KRWFAIF.E9260479 CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00413W >>>> VOLUME=MVS344 - SYSID=S008 31MAY91:21:01:50.70 - DATA SETS HAVE BEEN DELETED FROM VCA_VS
10.34.49 VCA00413W
10.34.49 VCA00415I CATALOG.MVSICF1.BACKUPS NO MATCHING VVDS RECORD
10.34.49 VCA00417I DATA (B) CATALOG.MVSICF1.BACKUPS CATALOG = CATALOG.MVSICF1.BACKUPS
10.34.49 VCA00415I CATALOG.MVSICF1.BACKUPS NO MATCHING VVDS RECORD
10.34.49 VCA00417I INDEX (B) CATALOG.MVSICF1.BACKUPS.CATINDEX CATALOG = CATALOG.MVSICF1.BACKUPS
10.34.49 VCA00415I DSICM11.TPX200.ADMIN1 NO MATCHING VVDS RECORD
10.34.49 VCA00417I DATA (B) DSICM11.TPX200.ADMIN1.DATA CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DSICM11.TPX200.ADMIN1 NO MATCHING VVDS RECORD
10.34.49 VCA00417I INDEX (B) DSICM11.TPX200.ADMIN1.IX CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DSICM11.TPX200.ADMIN2 NO MATCHING VVDS RECORD
10.34.49 VCA00417I DATA (B) DSICM11.TPX200.ADMIN2.DATA CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DSICM11.TPX200.ADMIN2 NO MATCHING VVDS RECORD
10.34.49 VCA00417I INDEX (B) DSICM11.TPX200.ADMIN2.IX CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DSICM11.TPX301.ADMIN2 NO MATCHING VVDS RECORD
10.34.49 VCA00417I DATA (B) DSICM11.TPX301.ADMIN2.DATA CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DSICM11.TPX301.ADMIN2 NO MATCHING VVDS RECORD
10.34.49 VCA00417I INDEX (B) DSICM11.TPX301.ADMIN2.INDEX CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DSIRE11.BUNDL.DPMFVIF NO MATCHING VVDS RECORD
10.34.49 VCA00417I DATA (B) DSIRE11.BUNDL.DPMFVIF.DATA CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DSIRE11.BUNDL.DPMFVIF NO MATCHING VVDS RECORD
10.34.49 VCA00417I INDEX (B) DSIRE11.BUNDL.DPMFVIF.INDEX CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DSIRF11.TPX2.ADMIN2 NO MATCHING VVDS RECORD
10.34.49 VCA00417I DATA (B) DSIRF11.TPX2.ADMIN2.DATA CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DSIRF11.TPX2.ADMIN2 NO MATCHING VVDS RECORD
10.34.49 VCA00417I INDEX (B) DSIRF11.TPX2.ADMIN2.IX CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DXB168.R521.CSI NO MATCHING VVDS RECORD
10.34.49 VCA00417I DATA (B) DXB168.R521.CSI.DATA CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DXB168.R521.CSI NO MATCHING VVDS RECORD
10.34.49 VCA00417I INDEX (B) DXB168.R521.CSI.INDEX CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DXB168.TEST.CSI NO MATCHING VVDS RECORD
10.34.49 VCA00417I DATA (B) DXB168.TEST.CSI.DATA CATALOG = CATALOG.MVSICF1.VMVSWX
10.34.49 VCA00415I DXB168.TEST.CSI NO MATCHING VVDS RECORD
10.34.49 VCA00417I INDEX (B) DXB168.TEST.CSI.INDEX CATALOG = CATALOG.MVSICF1.VMVSWX

```

Figure 9-5. VSAM Exception Report (Part 1 of 2)

10.34.49	VCA00415I		SA1PC78.TPX200.ADMIN1		NO MATCHING VVDS RECORD
10.34.49	VCA00417I	DATA (B)	SA1PC78.TPX200.ADMIN1.DATA		CATALOG = CATALOG.MVSICF1.VMVSXWA
10.34.49	VCA00415I		SA1PC78.TPX200.ADMIN1		NO MATCHING VVDS RECORD
10.34.49	VCA00417I	INDEX (B)	SA1PC78.TPX200.ADMIN1.INDEX		CATALOG = CATALOG.MVSICF1.VMVSXWA
10.34.49	VCA00415I		SA1PC78.TPX200.ADMIN2		NO MATCHING VVDS RECORD
10.34.49	VCA00417I	DATA (B)	SA1PC78.TPX200.ADMIN2.DATA		CATALOG = CATALOG.MVSICF1.VMVSXWA
10.34.49	VCA00415I		SA1PC78.TPX200.ADMIN2		NO MATCHING VVDS RECORD
10.34.49	VCA00417I	INDEX (B)	SA1PC78.TPX200.ADMIN2.INDEX		CATALOG = CATALOG.MVSICF1.VMVSXWA
10.34.49	VCA00415I		TPX.TPX30ACL.ADMIN1		NO MATCHING VVDS RECORD
10.34.49	VCA00417I	DATA (B)	TPX.TPX30ACL.ADMIN1.DATA		CATALOG = CATALOG.MVSICF1.VMVSXWA
10.34.49	VCA00415I		TPX.TPX30ACL.ADMIN1		NO MATCHING VVDS RECORD
10.34.49	VCA00417I	INDEX (B)	TPX.TPX30ACL.ADMIN1.INDEX		CATALOG = CATALOG.MVSICF1.VMVSXWA
10.34.49	VCA00415I		TPX.TPX30BT2.ADMIN1		NO MATCHING VVDS RECORD
10.34.49	VCA00417I	DATA (B)	TPX.TPX30BT2.ADMIN1.DATA		CATALOG = CATALOG.MVSICF1.VMVSXWA
10.34.49	VCA00415I		TPX.TPX30BT2.ADMIN1		NO MATCHING VVDS RECORD
10.34.49	VCA00417I	INDEX (B)	TPX.TPX30BT2.ADMIN1.INDEX		CATALOG = CATALOG.MVSICF1.VMVSXWA
10.34.52	VCA00413W	>>>>>	VOLUME=TSU021 - SYSID=S008 31MAY91:21:01:50.70 -		DATA SETS HAVE BEEN DELETED FROM VCA_VS
10.34.52	VCA00413W				
10.34.52	VCA00415I		ATM.DIST.AOI20.ADMIN2		NO MATCHING VVDS RECORD
10.34.52	VCA00417I	DATA (B)	ATM.DIST.AOI20.ADMIN2.DATA		CATALOG = CATALOG.MVSICF1.VMVSXWA
10.34.53	VCA00413W	>>>>>	VOLUME=TSU025 - SYSID=S008 31MAY91:21:01:50.70 -		DATA SETS HAVE BEEN DELETED FROM VCA_VS
10.34.53	VCA00413W				
10.34.53	VCA00415I		SYS1.VVDS.VTSU025		NO MATCHING VVDS RECORD
10.34.53	VCA00417I	DATA (B)	SYS1.VVDS.VTSU025		CATALOG = CATALOG.MVSICF1.BACKUPS
10.35.02	VCA00418W	>>>>>	29 VSAM DATA SETS DELETED FROM	5 VOLUMES.	

Figure 9-5. VSAM Exception Report (Part 2 of 2)

## 9.2.6 Phase 5 - Merge VSAM Information into VOA

In this phase, the information on VSAM data sets that was consolidated into the \_VS file is now used to update the Volume Activity (VOA) file. Each VSAM observation has its data summed into the statistics for a given volume. If any VSAM records are detected for a volume that is not in the work VOA file created in Phase 1 processing, the VSAM/Volume Exception Report is written to the MICSL0G Sysout Data Set.

For each observation that will be written to the output VOA file, the user exit, USRSVOA, is invoked. The final processing in this phase re-sorts the VOA file into its final order, and writes it to the CA MICS database.

### 9.2.7 Phase 6 - Match VOA to DAA

This phase begins by appending SMS data extracted during Phase 1 to that extracted during Phase 3. This SMS data is then sorted and merged with the VCAVOA, VCADAA, and EXTDAAs work files in order to create the detail level of the DAA file.

DSNODES 1 through 5 and DSNODEL are created for use by the VCAACRT Accounting routine. Various error conditions are checked, such as a missing DFSMS Information Cell, or a mismatch of VVDS Data to the VTOC. If any such errors are detected, the DFSMS Exception Report/VTOC Restart Report is written to the MICSLOG Sysout data set (see Figure 9-6 for an example of this report).

Checks are also made in this phase for DFHSM-owned structures in order to set the element DAADSTYP (data set type) to 'HS' or 'HV' and element DAACSTAT (Catalog Status) to 'HSM' so that duplicate billing for HSM occupied DASD space can be avoided. That is, the VCAACRT account routine can ignore this space if it is billed by the HSMACRT account routine.

The VCAACRT routine is invoked to cost each DAA observation as needed. The DFSMS storage group name obtained from the VOA file is propagated to each data set observation and the user exit USRSDAA is invoked.

If EXTENTDETAIL was requested for VSAM data sets having more than 16 extents (see Section 7.3.1), multiple observations are output describing each set of 16 extents. At the end of this phase, the work data sets are deleted and the final version of the DAA file is re-sorted and written to the CA MICS database.

```

----- MICSLOG ----- 19:00 SATURDAY, MAY 11, 1991
19.04.28 VCA00601I          DFSMS EXCEPTION REPORT / VTOC RESTART REPORT
19.04.28 VCA00601I
19.04.28 VCA00601I
19.04.28 VCA00601I          THIS REPORT IDENTIFIES SEVERAL POSSIBLE ERROR CONDITIONS WHICH STEM FROM PROBLEMS IN THE VCC
19.04.28 VCA00601I          COLLECTION PROCESS --- OR --- DFSMS FLAG INCONSISTENCIES AT THE VOLUME OR THE DATA SET LEVEL.
19.04.28 VCA00601I
19.04.28 VCA00601I          FOR MESSAGE GROUP VCA00602W, EITHER:
19.04.28 VCA00601I          (1) SMS MANAGED VOLUME CONTAINS DATA SETS WHICH ARE NOT SMS-MANAGED (MISSING SMS CELL), OR,
19.04.28 VCA00601I          (2) FORMAT-1 DSCB FIELDS RESERVED FOR USE BY DFSMS HAVE BEEN USED BY NON-IBM CODE RESULTING
19.04.28 VCA00601I          IN A FALSE INDICATION OF AN SMS-MANAGED DATA SET.
19.04.28 VCA00601I          ACTION: OBSERVATION IS WRITTEN WITH SMS DATA ELEMENTS MISSING.
19.04.28 VCA00601I
19.04.28 VCA00601I          FOR MESSAGE GROUP VCA00607W:
19.04.28 VCA00601I          (1) VCC DATA COLLECTION FOR THE SPECIFIED VOLUME DID NOT COMPLETE PROPERLY. THE CAUSE IS
19.04.28 VCA00601I          SHOWN IN THE VCCSTATS REPORT.
19.04.28 VCA00601I          ACTION: OBSERVATION IS DISCARDED FROM THE VCA00607W FILE (SINCE VCA00607W OBSERVATION IS MISSING).
19.04.28 VCA00601I  -- -----

19.04.28 VCA00602W >>>>> VOLUME=TSU020 - SYSID=PRD1 10MAY91:15:31:52.57 - ERRORS MATCHING VVDS (DFSMS CELL) DATA TO VTOC
19.04.28 VCA00602W
19.04.28 VCA00603W 05      TDKPE22.ASXJES3.LNK          DFSMS INFORMATION CELL MISSING
19.04.29 VCA00603W 05      IAM.IAMED.LOGDATA          DFSMS INFORMATION CELL MISSING
19.04.29 VCA00603W 05      SMSTEST.EDITEST          DFSMS INFORMATION CELL MISSING

19.04.29 VCA00602W >>>>> VOLUME=TSU021 - SYSID=PRD1 10MAY91:15:31:52.57 - ERRORS MATCHING VVDS (DFSMS CELL) DATA TO VTOC
19.04.29 VCA00602W
19.04.29 VCA00603W VS      ATM.DIST.AOI20.ADMIN2.DATA          DFSMS INFORMATION CELL MISSING
19.04.29 VCA00606W ??      TDKWFVA.SMPWRK3.T1822070.D91039.0BJ          MANAGED DATA SET IN VVDS BUT NOT PRESENT IN VTOC

19.04.30 VCA00602W >>>>> VOLUME=TSU022 - SYSID=PRD1 10MAY91:15:31:52.57 - ERRORS MATCHING VVDS (DFSMS CELL) DATA TO VTOC
19.04.30 VCA00602W
19.04.30 VCA00603W 05      DPO.QFS25.LOAD          DFSMS INFORMATION CELL MISSING
19.04.31 VCA00603W 05      TNG0807.ISPFXA.ISPPROF          DFSMS INFORMATION CELL MISSING

19.04.33 VCA00607W >>>>> VOLUME=TSU026 - SYSID=PRD1 10MAY91:15:31:52.57 - VOL. NOT SUCCESSFULLY PROCESSED
19.04.33 VCA00607W
19.04.33 VCA00606W ??      TDKDSUK.ISR0002.BACKUP          MANAGED DATA SET IN VVDS BUT NOT PRESENT IN VTOC

19.04.36 VCA00607W >>>>> VOLUME=TSU028 - SYSID=PRD1 10MAY91:15:31:52.57 - VOL. NOT SUCCESSFULLY PROCESSED
19.04.36 VCA00607W
19.04.36 VCA00606W ??      TDK179.HST9104C.TEMP1.CNTL          MANAGED DATA SET IN VVDS BUT NOT PRESENT IN VTOC

19.04.36 VCA00602W >>>>> VOLUME=TSU029 - SYSID=PRD1 10MAY91:15:31:52.57 - ERRORS MATCHING VVDS (DFSMS CELL) DATA TO VTOC
19.04.36 VCA00602W
19.04.36 VCA00603W 05      TDKMG11.SMF.DATA          DFSMS INFORMATION CELL MISSING
19.04.37 VCA00603W 05      TDK081.JCL.CNTL          DFSMS INFORMATION CELL MISSING

19.04.37 VCA00602W >>>>> VOLUME=TSU030 - SYSID=PRD1 10MAY91:15:31:52.57 - ERRORS MATCHING VVDS (DFSMS CELL) DATA TO VTOC
19.04.37 VCA00602W
19.04.37 VCA00603W 05      BACKUP.DATASETS.SETS1.LIST.G0059V00          DFSMS INFORMATION CELL MISSING
19.04.38 VCA00603W 05      TDKKR11.BUNTRACE          DFSMS INFORMATION CELL MISSING

19.04.55 VCA00608W >>>>>          0 DATA SETS IGNORED FROM          2 VOLUMES.

```

Figure 9-6. DAY090 MICSLOG DFSMS Exception / VTOC Restart Report

### 9.2.8 Phase 7 - Summarization and Aging

The final phase of the DAY090 step invokes the source program modules DYVCASUM and DYVCAAGE. DYVCASUM begins by deleting any 00 cycles in the DAY, WEEKS, and MONTHS timespans and also the highest cycle in the DAYS timespan. Summarization MACROs are invoked for each file, including the Data Class Summary (DCS) file, which is derived from the \_VS and DAA files. Since this file is not supported in the DETAIL timespan of the CA MICS database, the VCADCS00 cycle created during summarization is deleted.

In some cases, a special DAYS summarization routine is needed for special purposes such as:

- identifying unique ENDTS values enabling DAASAMPL to be set to 1 or to 0 prior to summarization.
- inserting a macro call to the costing routine generated by the CA MICS Accounting and Chargeback product when ACCOUNTING DAYS was specified in the VCA unit.

Therefore, the following routines are %INCLUDEd when their respective files are being summarized from DETAIL to DAYS:

- sharedprefix.MICS.SOURCE(#DAADSUM) for the VCADAA file
  - a call is made to the USRDDAA exit from this module
- sharedprefix.MICS.SOURCE(#0VSDSUM) for the VCA\_VS file
  - a call is made to the USRD\_VS exit from this module

DYVCAAGE updates the checkpoint file and logs an image of the checkpoint record to the MICSLOG Sysout data set. PROC DATASETS is invoked to age the DETAIL and DAYS 00 cycles to 01, 01 to 02, etc. For the MONTHS timespan the most current cycle created by summarization has an M0 suffix. The AGERNOT MACRO ages this cycle to become the updated month-to-date cycle (00) for each data set supported in that timespan.

## 9.3 Daily Processing Flow for HSM and HSB (DAY095)

This section describes the daily processing flow for the HSM and HSB Information Areas and its relation to the locations of the HSM user exit routines. For specific questions not discussed in this document, see the source code referenced by the program module.

The CA MICS HSM update process (the DAY095 step of the DAILY update job) is divided into three distinct processing phases. The following sections describe the general logic flow in each of the phases, as well as the location of standard user routines or user exits.

- 1 - MICSLOG Reporting
- 2 - Phase 1 - Input Raw VCC SMF Data
- 3 - Phase 2 - Sort and Build Files
- 4 - Phase 3 - Summarization and Aging
- 5 - Description of Phase 2 Error Handling

### 9.3.1 MICCSLOG Reporting

The MICCSLOG Sysout data set is used to display information on the input sources and progress of the Daily update step DAY095. An example of the MICCSLOG data set produced by a typical Daily update job is shown in Figure 9-7. The source of the input for this run was SYSID PRD1, and no records were excluded from the source file due to invalid record conditions. The progress of each processing phase is indicated by the message time stamps, and the checkpoint file is updated to reflect the data processed during the run.

Where the DAY090 step relies on the presence of a VCAVOA observation to signify that data collection was "complete" for a given volume, the DAY095 step processing DFHSM inventory data has to rely on the presence of a special inventory "trailer" record to signify a complete inventory for a given SYSID at a given collection time (ENDTS).

Figure 9-8 shows the MICCSLOG produced when incomplete data is processed. In this case, several "orphan" records (that is, records without a matching trailer record) were detected in the input data, and the system purged these records to avoid duplication that might occur in the processing of a subsequent VCC inventory run.

```

----- MICSLOG ----- 17:19 WEDNESDAY, MAY 15, 1991
17.19.44 BAS00119I AT CHECKPOINT FOR STEP "DAY095" IN JOB "DAILY" FOR DATABASE "VCAUNIT".
17.19.44 BAS00122I REQUIRED PREDECESSOR STEP IS:
17.19.44 BAS00122I     PROCESS=DAY
17.19.44 BAS00122I     STEP   =090
17.19.44 BAS00125I CHECKPOINT STEPS: D_STEP=090 W_STEP=900 M_STEP=900 Y_STEP=900 B_STEP=900 R_STEP=900
17.19.44 BAS00125I

17.21.17 BAS00318I ORGSYSID/COMPONENT CHECKPOINT INFORMATION BEFORE PROCESSING

17.21.17 BAS00346I  PRD1 VCA      10MAY91:15:31:52.57 10MAY91:15:31:52.57

17.21.17 BAS00346I  PRD1 HSM      10MAY91:13:28:14.81 10MAY91:13:28:14.81

17.21.18 HSM00100I ----- BEGIN PHASE 1 -----
17.21.18 HSM00011I HSMPTGEN PARAMETERS IN EFFECT:
17.21.18 HSM00011I >>>>> ACCOUNTING DETAIL

17.21.18 HSM00104I FIRST INPUT TIMESTAMP IS FROM SYSID (PRD1)
17.21.18 HSM00104I -- VERSION (1) DATA CREATED ON 14MAY91:18:25:14.12.
17.21.18 HSM00104I -- COMPUTED DURATION = 100:56:59.3
17.21.18 HSM00104I -- BASED ON A PREVIOUS COLLECTION ON 10MAY91:13:28:14.81

17.24.33 HSM00110I HSM INVENTORY TRAILER RECORD << FOUND >> FOR SYSID (PRD1)
17.24.33 HSM00110I -- VERSION (1) DATA CREATED ON 14MAY91:18:25:14.12.
17.24.33 HSM00111I   RECORD-TYPE CODE      COUNT          INCOMING
17.24.33 HSM00112I -- MIGRATE           M      21890          =      21890
17.24.33 HSM00112I -- BACKUP            B     48141          =      48141
17.24.33 HSM00112I -- DASD-CP           C       443           =       443
17.24.33 HSM00112I -- TAPE-CP           T         3           =         3

17.24.33 BAS00326I INPUT FILE(S) CONTAINED 70478 RECORDS FROM 14MAY1991:18:25:14.12 THRU 14MAY1991:18:25:14.12
17.24.33 BAS00326I
17.24.33 BAS00331I DATA SELECTION STATISTICS BY SYSTEM/COMPONENT
17.24.33 BAS00331I
17.24.33 BAS00331I   - - - - - DELETED - - - -
17.24.33 BAS00331I SYSTEM COMPONENT      INPUT  DATE RANGE  SHORT RECORDS  OPTION  KEPT
17.24.33 BAS00331I PRD1 HSM                70478      0           0           0       70478
17.24.33 BAS00343I TOTAL RECORDS KEPT = 70478
17.24.33 BAS00337I

17.24.36 HSM00200I ----- BEGIN PHASE 2 -----

17.28.41 HSM00999I ----- BEGIN SUMMARIZATION AND AGING -----
17.30.09 BAS00339I CHECKPOINT FILE ORGSYSID/COMPONENT UPDATE FOR COMPONENT HSM

17.30.09 BAS00340I BEFORE UPDATE: PRD1 HSM      10MAY91:13:28:14.81 10MAY91:13:28:14.81

17.30.09 BAS00341I AFTER UPDATE: PRD1 HSM      10MAY91:13:28:14.81 14MAY91:18:25:14.12
17.30.22 BAS00121I AT FINAL POST FOR STEP "DAY095" IN JOB "DAILY"

```

Figure 9-7. DAY095 MICSLOG Standard Output

### 9.3.2 Phase 1 - Input Raw VCC SMF Data

The Input Phase reads data records from the file defined by the INPUTSMF DD statement. The user exit USRSEL is executed if defined. The record type is checked and non-HSM records are ignored. The record length is checked to verify that the record is of the proper format. A check is also made to insure that the record was produced by VCC at the proper level.

During the input phase, the checkpoint file is checked to prevent data which has already been input to CA MICS from being input to CA MICS a second time. A check is also performed to ascertain if multiple inventory (VCC) runs are being input.

The main function of this first phase is the creation of a series of work files corresponding to the different subtypes that might appear in each record produced by VCC. Type 'M' and 'B' (Migration and Backup) records contain similar information, and are largely processed by the same code, but separate work files HSMIG00 and HSBAC00 are created. Prior to writing the HSMIG00 work file, the DSNODE 1 through 5 and DSNODEL data elements are built, and the HSMACRT Account Code Derivation Exit is called to assign account code values for each migrated data set processed.

Type 'B' records represent backup versions of data sets. Because there are likely multiple backup versions or copies of a given data set, processing time is saved by deferring the %INCLUDE of HSMACRT until the work version of the HSBAC00 file has been sorted and reduced to a single observation per data set name. Therefore, backup version records are simply written to a work data set in phase 1 in preparation for phase 2 processing.

The subtype 'C' records are used in creating the Volume Planning File (HSMVOL), and the input processing phase simply creates an observation for each record on the WORK.HSMVOL00 file.

The subtype 'T' records are used in creating the Tape Planning File (HSMTAP). Since separate records are created for each type of tape under DFHSM control (Backup, Dump and Migration), they get summarized into consolidated records during phase 2 processing. Consequently, the user output exit is not invoked for this file until that time.

A final record, subtype 'Z', marks the end of a single

inventory run. When this record is detected a single record is written to the WORK.SNAPSHOT file to indicate the SYSID and end timestamp of a complete inventory. Informational messages are also written to the MICSLOG Sysout data set indicating the number of records processed in the inventory.

Once all of the input data has been processed, control is passed to the next phase for sorting of the created work files and further processing.

### 9.3.3 Phase 2 - Sort and Build Files

This phase begins by checking to see if multiple inventories (unique SYSID and ENDTS) were present in the input.

If YES, then a flag is set for passing to the DAYS summarization routines (#fffDSUM) which signals that the data from the DETAIL timespan will need to be SORTED.

If NO, then the DAYS summarization routines may be able to skip the SORTing of the DETAIL data into the DAYS timespan sequence. This status check is an indicator of possible performance improvements; refer to Appendix A for an explanation of message HSM00300I.

In general, phase 2 processing aims to accomplish the following objectives:

1. PROC SORT with the NODUP option on all WORK files
  - a. Sufficient granularity is introduced by adding such extra fields such as MIGRATTS (time of migrate or time of backup) to arrange any duplicate observations next to each other so PROC SORT can purge the duplicates automatically.
2. Final processing prior to updating DETAIL timespan. Each file has slightly different requirements.
  - a. Backup versions are summarized to the DSNAME level (aggregate all versions of a given data set name using normal CA MICS summarization macros).

Once reduced to a single observation per data set, then the following exit points are called:

- DSNODEs are parsed from the full DSNAME
- HSMACRT is %INCLUDEd to assign account codes
- The cost routine generated by CA MICS Accounting and Chargeback is called to price the current inventory of backup versions (journal file for backup resources will only be created if ACCOUNTING DETAIL is in effect)
- The USRSBAC exit is called to allow ultimate user control over content and existence of this observation.

Once built in this format, the HSBAC00 file is sorted again into the final sequence dictated by the DETAIL

timespan. If the DETAIL timespan is inactive to save space, it is left on a WORK file for later absorption into the DAYS timespan.

- b. Migrated data set observations have already been assigned account codes in phase 1. Therefore:
  - The cost routine generated by CA MICS Accounting and Chargeback is called to price the migrated data set (again, only if ACCOUNTING DETAIL is in effect).
  - The USRSMIG exit is called to allow ultimate user control over content and existence of this observation.

The HSMIG00 file is already sorted into the final sequence dictated by the DETAIL timespan and thus does not need to be sorted again. If the DETAIL timespan is inactive to save space, it is left on a WORK file for later absorption into the DAYS timespan.

- c. DASD planning observations do not have account codes and are already in their final DETAIL sequence and so:
  - The USRSVOL exit is called to allow ultimate user control over content and existence of this observation.
- d. Tape planning observations do not have account codes and are already in their final DETAIL sequence and so:
  - The USRSTAP exit is called to allow ultimate user control over content and existence of this observation.

### 9.3.4 Phase 3 - Summarization and Aging

The final phase of the DAY095 step invokes the source program modules DYHMSUM and DYHSMAGE. DYHMSUM begins by deleting any 00 cycles in the DAYS, WEEKS, and MONTHS timespans and also the highest cycle in the DAYS timespan. Summarization MACROs are the invoked for each file.

The FILE statements in HSMGENIN show an E for the DAYS timespan which means a special summarization module is to be called. The HSMTAP file is the only file with "standard" summarization but the other files get summarized with the following modules controlling the process:

a. sharedprefix.MICS.SOURCE(#BACDSUM)

- If possible, DETAIL data is not sorted into the DAYS sequence. Only if multiple inventories are present or if DAYS sort sequence is not contained as a subset of DETAIL will we need to sort the file again. See message HSM00300 in Appendix A.
- The cost routine generated by CA MICS Accounting and Chargeback is called to price the group of backup data sets summarized to a given account code structure (only if ACCOUNTING DAYS is in effect).
- The USRDBAC exit is called to allow ultimate user control over content and existence of this observation.

b. sharedprefix.MICS.SOURCE(#MIGDSUM)

- If possible, DETAIL data is not sorted into the DAYS sequence. Only if multiple inventories are present or if DAYS sort sequence is not contained as a subset of DETAIL will we need to SORT the file again. See message HSM00300 in Appendix A.
- The cost routine generated by the CA MICS Accounting and Chargeback product is called to price the group of migrated data sets summarized to a given account code structure (only if ACCOUNTING DAYS is in effect).
- The USRDMIG exit is called to allow ultimate user control over content and existence of this observation.

c. sharedprefix.MICS.SOURCE(#VOLDSUM)

- If possible, DETAIL data is not sorted into the DAYS

sequence. Only if multiple inventories are present or if DAYS sort sequence is not contained as a subset of DETAIL will we need to SORT the file again. See message HSM00300 in Appendix A.

- The USRDVOL exit is called to allow ultimate user control over content and existence of this observation.

DYVCAAGE updates the checkpoint file and logs an image of the checkpoint record to the MICSLOG Sysout data set. PROC DATASETS is invoked to age the DETAIL and DAYS 00 cycles to 01, 01 to 02, etc. For the WEEKS and MONTHS timespans the most current cycle created by summarization has a W0 or M0 suffix respectively. The AGERNOT MACRO ages this cycle to become the updated to-date cycle (00) for each data set supported in the specified timespan.

### 9.3.5 Description of Phase 2 Error Handling

Figure 9-8 is included to illustrate how the DAY095 step logs out error information to the MICSLOG data set. Fragments of Figure 9-8 are repeated here to better show the context of the error.

- a. The first record encountered in the input shows that the inventory was taken on 10 May 91. Message HSM00104I also shows the DURATION that was computed as a result of the previous inventory being taken at 12:58 on the same day.

```
18.44.21 HSM00104I FIRST INPUT TIMESTAMP IS FROM SYSID (PRD1)
18.44.21 HSM00104I -- VERSION (1) DATA CREATED ON 10MAY91:13:28:14.81.
18.44.21 HSM00104I -- COMPUTED DURATION = 0:30:01.43
18.44.21 HSM00104I -- BASED ON A PREVIOUS COLLECTION ON 10MAY91:12:58:13.38
```

- b. Then the subtype 'Z' record is found and the records counts contained within the record are shown under the column named "COUNT" in message HSM00112I. The column named "INCOMING" shows the result of counting the records by subtype in this DATA step. The incoming count should always equal the count produced by VCC and placed in the trailer record and in this example they are equal (this is shown by the equal sign = appearing between the two columns).

```
18.44.52 HSM00110I HSM INVENTORY TRAILER RECORD << FOUND >> FOR SYSID (PRD1)
18.44.52 HSM00110I -- VERSION (1) DATA CREATED ON 10MAY91:13:28:14.81.
18.44.52 HSM00111I RECORD-TYPE CODE COUNT INCOMING
18.44.52 HSM00112I -- MIGRATE M 21980 = 21980
18.44.52 HSM00112I -- BACKUP B 44985 = 44985
18.44.52 HSM00112I -- DASD-CP C 582 = 582
18.44.52 HSM00112I -- TAPE-CP T 3 = 3
```

- c. The next cluster of HSM00104I messages is followed by HSM00116I messages. Here a set of "orphan" records created on 11 May, 12 May, and 13 May has been found during the reading of the input data set. As HSM00116I states, these records will be discarded. DAY095 operates on the design principle that a complete inventory of data must be presented to CA MICS at one time. Partial inventories are discarded.

```

18.44.52 HSM00104I FIRST INPUT TIMESTAMP IS FROM SYSID (PRD1)
18.44.52 HSM00104I -- VERSION (1) DATA CREATED ON 11MAY91:13:28:14.81.
18.44.52 HSM00104I -- COMPUTED DURATION = 24:30:01.43
18.44.52 HSM00104I -- BASED ON A PREVIOUS COLLECTION ON 10MAY91:12:58:13.38

```

```

18.44.52 HSM00110I HSM INVENTORY TRAILER RECORD <<MISSING>> FOR SYSID (PRD1)
18.44.52 HSM00110I -- VERSION (1) DATA CREATED ON 11MAY91:13:28:14.81.
18.44.52 HSM00116I DATA WITH THIS ENDTS WILL BE PURGED TO AVOID DUPLICATION
18.44.52 HSM00116I -- BY A LATER RERUN OF A COMPLETED INVENTORY.

```

```

18.44.52 HSM00104I FIRST INPUT TIMESTAMP IS FROM SYSID (PRD1)
18.44.52 HSM00104I -- VERSION (1) DATA CREATED ON 12MAY91:13:28:14.81.
18.44.52 HSM00104I -- COMPUTED DURATION = 48:30:01.43
18.44.52 HSM00104I -- BASED ON A PREVIOUS COLLECTION ON 10MAY91:12:58:13.38

```

```

18.44.52 HSM00110I HSM INVENTORY TRAILER RECORD <<MISSING>> FOR SYSID (PRD1)
18.44.52 HSM00110I -- VERSION (1) DATA CREATED ON 12MAY91:13:28:14.81.
18.44.52 HSM00116I DATA WITH THIS ENDTS WILL BE PURGED TO AVOID DUPLICATION
18.44.52 HSM00116I -- BY A LATER RERUN OF A COMPLETED INVENTORY.

```

```

18.44.52 HSM00104I FIRST INPUT TIMESTAMP IS FROM SYSID (PRD1)
18.44.52 HSM00104I -- VERSION (1) DATA CREATED ON 13MAY91:13:28:14.81.
18.44.52 HSM00104I -- COMPUTED DURATION = 72:30:01.43
18.44.52 HSM00104I -- BASED ON A PREVIOUS COLLECTION ON 10MAY91:12:58:13.38

```

- d. Another possible error is shown below where an inventory trailer record (subtype 'Z') is found to have different internal record counts than what was encountered in the input data set. Rather than an equal sign, we highlight this mismatch with the message fragment that states: "WARNING: UNMATCHED" between the columns of count values.

```

----- MICSLOG ----- 18:43 TUESDAY, MAY 14, 1991
18.44.52 HSM00110I HSM INVENTORY TRAILER RECORD << FOUND >> FOR SYSID (PRD1)
18.44.52 HSM00110I -- VERSION (1) DATA CREATED ON 13MAY91:13:28:14.81.
18.44.52 HSM00111I RECORD-TYPE CODE COUNT INCOMING
18.44.52 HSM00112I -- MIGRATE M 21980 WARNING: UNMATCHED 6
18.44.52 HSM00112I -- BACKUP B 44985 WARNING: UNMATCHED 6
18.44.52 HSM00112I -- DASD-CP C 582 WARNING: UNMATCHED 0
18.44.52 HSM00112I -- TAPE-CP T 3 WARNING: UNMATCHED 4

```

- e. After the HSM00200I message ("BEGIN PHASE 2"), the next cluster of messages shows what is being discarded. The message text shows a variable from the observation that best identifies what is being discarded. For example, for MIGRATE or BACKUP files, the DSNAME is present and that best identifies the data being discarded so that content is shown in message HSM00207I. The count(s) of observations deleted from the WORK files is shown in

message HSM00209I.

```
----- MICSLOG ----- 18:43 TUESDAY, MAY 14, 1991
18.44.54 HSM00200I ----- BEGIN PHASE 2 -----
18.45.32 HSM00205I FOLLOWING RECORDS PURGED FROM (HSBBAC) FILE - SYSID (PRD1)
18.45.32 HSM00205I -- INVENTORY FROM 11MAY91:13:28:14.81 IS BEING DISCARDED.
18.45.32 HSM00206I D A T A   S E T   N A M E
18.45.32 HSM00207I ACCUMAX.DB2MSTG1.BASE
18.45.32 HSM00207I TSX182.MICS.DBD.USER.LOAD
18.45.32 HSM00207I TEST.MICSL.CPLX.LOCALMOD.CNTL

18.45.32 HSM00209I >>>>>      3 RECORDS DELETED FROM HSBBAC.

18.45.32 HSM00205I FOLLOWING RECORDS PURGED FROM (HSBBAC) FILE - SYSID (PRD1)
18.45.32 HSM00205I -- INVENTORY FROM 12MAY91:13:28:14.81 IS BEING DISCARDED.
18.45.32 HSM00206I D A T A   S E T   N A M E
18.45.32 HSM00207I ACCUMAX.DB2MSTG1.BASE
18.45.32 HSM00207I TSX182.MICS.DBD.USER.LOAD
18.45.32 HSM00207I TEST.MICSL.CPLX.LOCALMOD.CNTL

18.45.32 HSM00209I >>>>>      3 RECORDS DELETED FROM HSBBAC.
```

f. The next section illustrates an aspect of DAY095 processing that is not related to the error processing above. Message HSM00300I can be produced when errors have been found in the incoming data (as in this example) or when a perfectly acceptable inventory is found. HSM00300I is an informational message produced at the beginning of the summarization processing. It signals that the DETAIL data had to be sorted an extra time to prepare it properly for the summarization routines. If you process multiple inventories in a single DAY095 step, a sort will always be required to arrange the inventories together rather than interleaved (ENDTS is the rightmost in the SORT sequence). The other reason why an additional SORT is required is if you have done file tailoring of your HSMGENIN member and caused the DAYS sequence to no longer be a logical subset of the DETAIL sequence.

The absence of HSM00300I indicates better performance because the DETAIL sequence was sufficient for summary processing and the additional sort could be avoided. The distributed version of HSMGENIN should result in the absence of the HSM00300I message.

18.47.37 HSM00999I ----- BEGIN SUMMARIZATION AND AGING -----  
18.47.44 HSM00300I SORTING DETAIL DATA INTO DAYS TIMESPAN SEQUENCE FOR FILE:  
18.47.44 HSM00300I -- DETAIL.HSBAC00  
18.47.44 HSM00300I -- (EITHER) DAYS SEQUENCE IS NOT A SUBSET OF DETAIL.  
18.47.44 HSM00300I -- (OR) MULTIPLE ENDTS VALUES ARE PRESENT IN THE INPUT DATA.

18.48.05 HSM00300I SORTING DETAIL DATA INTO DAYS TIMESPAN SEQUENCE FOR FILE:  
18.48.05 HSM00300I -- DETAIL.HSMIG00  
18.48.05 HSM00300I -- (EITHER) DAYS SEQUENCE IS NOT A SUBSET OF DETAIL.  
18.48.05 HSM00300I -- (OR) MULTIPLE ENDTS VALUES ARE PRESENT IN THE INPUT DATA.

18.48.35 HSM00300I SORTING DETAIL DATA INTO DAYS TIMESPAN SEQUENCE FOR FILE:  
18.48.35 HSM00300I -- WORK1.HSMVOL00  
18.48.35 HSM00300I -- (EITHER) DAYS SEQUENCE IS NOT A SUBSET OF DETAIL.  
18.48.35 HSM00300I -- (OR) MULTIPLE ENDTS VALUES ARE PRESENT IN THE INPUT DATA.

```
----- MICSLOG ----- 18:43 TUESDAY, MAY 14, 1991
18.43.49 BAS00119I AT CHECKPOINT FOR STEP "DAY095" IN JOB "DAILY" FOR DATABASE "VCAUNIT".
18.43.49 BAS00122I REQUIRED PREDECESSOR STEP IS:
18.43.49 BAS00122I          PROCESS=DAY
18.43.49 BAS00122I          STEP   =090
18.43.49 BAS00125I CHECKPOINT STEPS: D_STEP=090 W_STEP=900 M_STEP=900 Y_STEP=900 B_STEP=900
R_STEP=900
18.43.49 BAS00125I

18.44.20 BAS00318I ORGSYSID/COMPONENT CHECKPOINT INFORMATION BEFORE PROCESSING
18.44.20 BAS00318I

18.44.21 HSM00100I ----- BEGIN PHASE 1 -----
18.44.21 HSM00011I HSMPGEN PARAMETERS IN EFFECT:
18.44.21 HSM00011I >>>>> ACCOUNTING DETAIL

18.44.21 HSM00104I FIRST INPUT TIMESTAMP IS FROM SYSID (PRD1)
18.44.21 HSM00104I -- VERSION (1) DATA CREATED ON 10MAY91:13:28:14.81.
18.44.21 HSM00104I -- COMPUTED DURATION = 0:30:01.43
18.44.21 HSM00104I -- BASED ON A PREVIOUS COLLECTION ON 10MAY91:12:58:13.38

18.44.52 HSM00110I HSM INVENTORY TRAILER RECORD << FOUND >> FOR SYSID (PRD1)
18.44.52 HSM00110I -- VERSION (1) DATA CREATED ON 10MAY91:13:28:14.81.
18.44.52 HSM00111I   RECORD-TYPE CODE      COUNT                INCOMING
18.44.52 HSM00112I -- MIGRATE          M      21980                =      21980
18.44.52 HSM00112I -- BACKUP           B      44985                =      44985
18.44.52 HSM00112I -- DASD-CP          C         582                =         582
18.44.52 HSM00112I -- TAPE-CP          T         3                  =         3

18.44.52 HSM00104I FIRST INPUT TIMESTAMP IS FROM SYSID (PRD1)
18.44.52 HSM00104I -- VERSION (1) DATA CREATED ON 11MAY91:13:28:14.81.
18.44.52 HSM00104I -- COMPUTED DURATION = 24:30:01.43
18.44.52 HSM00104I -- BASED ON A PREVIOUS COLLECTION ON 10MAY91:12:58:13.38

18.44.52 HSM00110I HSM INVENTORY TRAILER RECORD <<MISSING>> FOR SYSID (PRD1)
18.44.52 HSM00110I -- VERSION (1) DATA CREATED ON 11MAY91:13:28:14.81.
18.44.52 HSM00116I DATA WITH THIS ENDTs WILL BE PURGED TO AVOID DUPLICATION
18.44.52 HSM00116I -- BY A LATER RERUN OF A COMPLETED INVENTORY.

18.44.52 HSM00104I FIRST INPUT TIMESTAMP IS FROM SYSID (PRD1)
18.44.52 HSM00104I -- VERSION (1) DATA CREATED ON 12MAY91:13:28:14.81.
18.44.52 HSM00104I -- COMPUTED DURATION = 48:30:01.43
18.44.52 HSM00104I -- BASED ON A PREVIOUS COLLECTION ON 10MAY91:12:58:13.38

18.44.52 HSM00110I HSM INVENTORY TRAILER RECORD <<MISSING>> FOR SYSID (PRD1)
18.44.52 HSM00110I -- VERSION (1) DATA CREATED ON 12MAY91:13:28:14.81.
18.44.52 HSM00116I DATA WITH THIS ENDTs WILL BE PURGED TO AVOID DUPLICATION
18.44.52 HSM00116I -- BY A LATER RERUN OF A COMPLETED INVENTORY.
```

```

18.44.52 HSM00104I FIRST INPUT TIMESTAMP IS FROM SYSID (PRD1)
18.44.52 HSM00104I -- VERSION (1) DATA CREATED ON 13MAY91:13:28:14.81.
18.44.52 HSM00104I -- COMPUTED DURATION = 72:30:01.43
18.44.52 HSM00104I -- BASED ON A PREVIOUS COLLECTION ON 10MAY91:12:58:13.38

```

Figure 9-8. DAY095 MICSL0G Error Report Output (Part 1 of 4)

```

----- MICSL0G ----- 18:43 TUESDAY, MAY 14, 1991
18.44.52 HSM00110I HSM INVENTORY TRAILER RECORD << FOUND >> FOR SYSID (PRD1)
18.44.52 HSM00110I -- VERSION (1) DATA CREATED ON 13MAY91:13:28:14.81.
18.44.52 HSM00111I RECORD-TYPE CODE COUNT INCOMING
18.44.52 HSM00112I -- MIGRATE M 21980 WARNING: UNMATCHED 6
18.44.52 HSM00112I -- BACKUP B 44985 WARNING: UNMATCHED 6
18.44.52 HSM00112I -- DASD-CP C 582 WARNING: UNMATCHED 0
18.44.52 HSM00112I -- TAPE-CP T 3 WARNING: UNMATCHED 4

18.44.52 BAS00326I INPUT FILE(S) CONTAINED 67568 RECORDS FROM 10MAY1991:13:28:14.81 THRU
10MAY1991:13:28:14.81
18.44.52 BAS00326I
18.44.52 BAS00326I
18.44.52 BAS00331I DATA SELECTION STATISTICS BY SYSTEM/COMPONENT
18.44.52 BAS00331I
18.44.52 BAS00331I - - - - - DELETED - - - -
18.44.52 BAS00331I SYSTEM COMPONENT INPUT DATE RANGE SHORT RECORDS OPTION KEPT
18.44.52 BAS00331I
18.44.52 BAS00343I PRD1 HSM 67568 0 0 0 67568

18.44.52 BAS00337I TOTAL RECORDS KEPT = 67568

```

Figure 9-8. DAY095 MICSL0G Error Report Output (Part 2 of 4)

```
----- MICSLOG ----- 18:43 TUESDAY, MAY 14, 1991
18.44.54 HSM00200I ----- BEGIN PHASE 2 -----
18.45.32 HSM00205I FOLLOWING RECORDS PURGED FROM (HSBBAC) FILE - SYSID (PRD1)
18.45.32 HSM00205I -- INVENTORY FROM 11MAY91:13:28:14.81 IS BEING DISCARDED.
18.45.32 HSM00206I D A T A   S E T   N A M E
18.45.32 HSM00207I ACCUMAX.DB2MSTG1.BASE
18.45.32 HSM00207I TSX182.MICS.DBD.USER.LOAD
18.45.32 HSM00207I TEST.MICSL.CPLX.LOCALMOD.CNTL

18.45.32 HSM00209I >>>>>      3 RECORDS DELETED FROM HSBBAC.

18.45.32 HSM00205I FOLLOWING RECORDS PURGED FROM (HSBBAC) FILE - SYSID (PRD1)
18.45.32 HSM00205I -- INVENTORY FROM 12MAY91:13:28:14.81 IS BEING DISCARDED.
18.45.32 HSM00206I D A T A   S E T   N A M E
18.45.32 HSM00207I ACCUMAX.DB2MSTG1.BASE
18.45.32 HSM00207I TSX182.MICS.DBD.USER.LOAD
18.45.32 HSM00207I TEST.MICSL.CPLX.LOCALMOD.CNTL

18.45.32 HSM00209I >>>>>      3 RECORDS DELETED FROM HSBBAC.

18.45.32 HSM00205I FOLLOWING RECORDS PURGED FROM (HSBBAC) FILE - SYSID (PRD1)
18.45.32 HSM00205I -- INVENTORY FROM 13MAY91:13:28:14.81 IS BEING DISCARDED.
18.45.32 HSM00206I D A T A   S E T   N A M E
18.45.32 HSM00207I

18.45.32 HSM00209I >>>>>      1 RECORDS DELETED FROM HSBBAC.

18.47.10 HSM00205I FOLLOWING RECORDS PURGED FROM (HSMVOL) FILE - SYSID (PRD1)
18.47.10 HSM00205I -- INVENTORY FROM 13MAY91:13:28:14.81 IS BEING DISCARDED.
18.47.10 HSM00216I VOLSER  COLLECTION-DATE
18.47.10 HSM00217I      .

18.47.10 HSM00209I >>>>>      1 RECORDS DELETED FROM HSMVOL.
18.47.17 HSM00205I FOLLOWING RECORDS PURGED FROM (HSMTAP) FILE - SYSID (PRD1)
18.47.17 HSM00205I -- INVENTORY FROM 11MAY91:13:28:14.81 IS BEING DISCARDED.

18.47.17 HSM00209I >>>>>      2 RECORDS DELETED FROM HSMTAP.
18.47.17 HSM00205I FOLLOWING RECORDS PURGED FROM (HSMTAP) FILE - SYSID (PRD1)
18.47.17 HSM00205I -- INVENTORY FROM 12MAY91:13:28:14.81 IS BEING DISCARDED.

18.47.17 HSM00209I >>>>>      2 RECORDS DELETED FROM HSMTAP.
18.47.17 HSM00205I FOLLOWING RECORDS PURGED FROM (HSMTAP) FILE - SYSID (PRD1)
18.47.17 HSM00205I -- INVENTORY FROM 13MAY91:13:28:14.81 IS BEING DISCARDED.

18.47.17 HSM00209I >>>>>      1 RECORDS DELETED FROM HSMTAP.

18.47.37 HSM00999I ----- BEGIN SUMMARIZATION AND AGING -----
18.47.44 HSM00300I SORTING DETAIL DATA INTO DAYS TIMESPAN SEQUENCE FOR FILE:
18.47.44 HSM00300I -- DETAIL.HSBBAC00
```

```
18.47.44 HSM00300I -- (EITHER) DAYS SEQUENCE IS NOT A SUBSET OF DETAIL.  
18.47.44 HSM00300I -- (OR) MULTIPLE ENDTS VALUES ARE PRESENT IN THE INPUT DATA.  
  
18.48.05 HSM00300I SORTING DETAIL DATA INTO DAYS TIMESPAN SEQUENCE FOR FILE:
```

Figure 9-8. DAY095 MICSL0G Error Report Output (Part 3 of 4)

```
----- MICSL0G ----- 18:43 TUESDAY, MAY 14, 1991  
18.48.05 HSM00300I -- DETAIL.HSM00300  
18.48.05 HSM00300I -- (EITHER) DAYS SEQUENCE IS NOT A SUBSET OF DETAIL.  
18.48.05 HSM00300I -- (OR) MULTIPLE ENDTS VALUES ARE PRESENT IN THE INPUT DATA.  
  
18.48.35 HSM00300I SORTING DETAIL DATA INTO DAYS TIMESPAN SEQUENCE FOR FILE:  
18.48.35 HSM00300I -- WORK1.HSMVOL00  
18.48.35 HSM00300I -- (EITHER) DAYS SEQUENCE IS NOT A SUBSET OF DETAIL.  
18.48.35 HSM00300I -- (OR) MULTIPLE ENDTS VALUES ARE PRESENT IN THE INPUT DATA.  
  
18.48.48 BAS00339I CHECKPOINT FILE ORGSYSID/COMPONENT UPDATE FOR COMPONENT HSM  
18.48.48 BAS00340I BEFORE UPDATE:  
  
18.48.48 BAS00341I AFTER UPDATE: PRD1 HSM 10MAY91:13:28:14.81 10MAY91:13:28:14.81  
  
18.48.48 BAS00341I  
18.48.55 BAS00121I AT FINAL POST FOR STEP "DAY095" IN JOB "DAILY"
```

Figure 9-8. DAY095 MICSL0G Error Report Output (Part 4 of 4)



# Chapter 10: MODIFICATION

---

The CA MICS Space Analyzer is tailored primarily through the use of options and parameters, file and data element tailoring, and exits. This chapter provides considerations for changing the product's parameters and describes the exits that are available during the DAILY update processing.

NOTE: Because this product has two operational components (VCA and HSM), there are aspects in common to each that are most effectively described in general terms by referring to both components by "ccc". For example, whenever you find a reference to cccCGEN, you should assume both VCACGEN and HSMCGEN (if applicable in your configuration).

This section contains the following topics:

- [10.1 Parameter Modification](#) (see page 337)
- [10.2 Standard User Exits](#) (see page 347)
- [10.3 File and Data Element Customization](#) (see page 367)
- [10.4 Testing Modifications to the Analyzer](#) (see page 368)
- [10.5 A Case Study Using RACF and DMS/OS Data](#) (see page 368)
- [10.6 Change the Number of Work Files for DAY090](#) (see page 379)
- [10.7 Change the Number of Work Files for DAY095](#) (see page 382)
- [10.8 Enable Internal Step Restart](#) (see page 385)
- [10.9 Implement Incremental Update](#) (see page 387)

## 10.1 Parameter Modification

Numerous parameters control the operation of the CA MICS Space Analyzer. Often, you can customize the product by altering the same parameters that you used to install the product. This section provides guidelines for customizing the CA MICS Space Analyzer through parameter modification. The following topics are addressed:

- 1 - Add or Delete a System in a Unit
- 2 - Add or Change Elements in a File
- 3 - Alter Derived Element Support
- 4 - Change Active Timespans for an Element
- 5 - Alter the Active Timespans for a File
- 6 - Add a New File
- 7 - Change the Number of Work Files for DAYnnn

### 10.1.1 Add or Delete a System in a Unit

If you are adding a system (a SYSID with CA MICS Space Collector data), to a unit, there are certain requirements imposed by the CA MICS base architecture that may need to be performed once.

#### **Scenario 1: SMF/RMF Already Defined in Unit**

Let's assume you already have a production unit database that processes SMF and RMF data from two MVS systems -- that is, from hypothetical SYSIDs defined as S001 and S002. You are planning the implementation of the CA MICS DASD products to collect and process data from either S001 or S002 (or both -- refer to the CA MICS Space Collector Guide for Shared DASD considerations).

Because you have already defined S001 and S002 as systems to be controlled in the unit, you can proceed directly to the parameter generation process for VCAPGEN and HSMPGEN as discussed in Chapter 7 of this guide.

#### **Scenario 2: DASD Data from Remote SYSID**

To build on scenario 1 above, let's now assume you want to process some VCC data that was collected from a new data center you have acquired. For discussion purposes, let's refer to this "remote" data center by its SYSID - R003.

Unlike scenario 1 where CA MICS Base software already "knew" about S001 and S002 as systems to be processed in this unit, you must define the SYSID called R003 and you must perform some housekeeping steps to satisfy the validation functions of CA MICS Base.

- a. Update prefix.MICS.PARMS(SYSID) with the new SYSID.

Section 2.3.2.2 in the PIOM describes this process.

- b. Update prefix.MICS.PARMS(PRFGP) with a dummy performance group for the new SYSID.

Section 2.3.2.5 in the PIOM describes this process.

- c. Submit the BASPGEN job in prefix.MICS.CNTL.

Section 3.5.5 in the PIOM describes this process.

After you complete these steps, you can proceed to the parameter generation processes for VCA and HSM described in detail in Section 7.3 of this guide.

### **Scenario 3: Deleting a SYSID**

To remove a system from the CA MICS Space Analyzer processing within a unit, you must determine if you are reversing the situation described in scenarios 1 or 2 above. Continuing with our hypothetical SYSIDs, you need to determine if any CA MICS product will be processing data with the SYSID value:

- a. If R003 is to be removed, simply delete the lines in members SYSID and PRFGP in prefix.MICS.PARMS that define R003, and run another BASPGEN job.

You may need to revisit Section 7.3 and delete data set references to data from R001 in prefix.MICS.PARMS member INPUTVCA and/or INPUTHSM.

- b. If S001 and/or S002 will no longer be generating VCC data (for example, perhaps a third SYSID called S003 has been added to scan all common DASD only), then you should do nothing because the assumption is S001 and S002 will still give off SMF and RMF data that needs to be processed and therefore, the SYSID and PRFGP definitions must remain intact.

## 10.1.2 Add or Change Elements in a File

You can add new elements to an existing file in the CA MICS Space Analyzer by updating:

- o `sharedprefix.MICS.GENLIB(VCAGENIN)`, or
- o `sharedprefix.MICS.GENLIB(HSMGENIN)`

Add NAME or NAMX statements (and TYPE statements, if required) after the FILE and FOPT statements for the file you are changing. After you make the change to the NAME, NAMX, or TYPE statement, you must run either:

- o `sharedprefix.MICS.CNTL(VCACGEN)`, or
- o `sharedprefix.MICS.CNTL(HSMCGEN)`

Refer to Section 4.2.4.3 of the System Modification Guide for more information about these statements.

## 10.1.3 Alter Derived Element Support

You can keep derived or computed elements (such as averages and percents) in the data base or computed when the elements are required. Computing these elements only when they are required can reduce the amount of DASD needed for the data base. The statements in `cccGENIN` that affect this option are `COMPOPT` and `FOPT`. These override the defaults set in `CPLXDEF`. After modifying these options in `cccGENIN`, you must run `cccCGEN`.

Refer to Sections 4.2.4.1.3 and 4.2.4.2.2 of the System Modification Guide for information on `COMPOPT` and `FOPT`.

Refer to Section 2.3.1.8 of the Planning, Installation, Operation, and Maintenance Guide (PIOM) for information on `CPLXDEF`.

## 10.1.4 Change Active Timespans For An Element

Alter the active timespans for an element by updating the element NAME or NAMX statement in the cccGENIN member of sharedprefix.MICS.PARMS. You can turn off an element completely and thereby remove it from the CA MICS Space Analyzer. This can be a useful way to reduce the size of your database.

Be very careful when altering the active timespans for an element, because it is possible to eliminate data from higher timespans (the element will become a missing value) if only the DETAIL or DAYS timespan is inactivated. After you make the change to the NAME or NAMX statement, you must run VCACGEN.

Refer to Section 4.2.4.3 of the System Modification Guide for more information about these statements.

Example: Turning Off Detailed Extent Information

Chapter 7 discusses a VCAPGEN parameter called EXTENTDETAIL which defaults to NO. This controls the "vertical" dimension of the VCADAA and VCA\_VS files by eliminating the "extra" observations that would be created by VCA to represent the detail extent locations for data sets having greater than 16 extents.

Let's assume you are looking for ways to conserve the amount of space taken by VCA and want to affect the "horizontal" dimension of the observations with regard to detail extent information.

The modification below will save a significant amount of space and the only tradeoff is that you will not be able to run the TRACKMAP reports (see Chapter 3 of this guide).

- a. Update sharedprefix.MICS.GENLIB(VCAGENIN) by changing the specification for the DETAIL timespan from 0 to N for everything but the first extent.

As distributed:

```
-----
NAME DAAXSZ01 00 0 N N N N
NAME DAAXSZ02 00 0 N N N N
NAME DAAXSZ03 00 0 N N N N
...(repeat for extents 04 through 16)...
NAME DAAXTK01 00 0 N N N N
NAME DAAXTK02 00 0 N N N N
```

```
NAME DAAXTK03 00 0 N N N N
...(repeat for extents 04 through 16)....
```

After your modification:

```
-----
NAME DAAXSZ01 00 0 N N N N
NAME DAAXSZ02 00 N N N N N
NAME DAAXSZ03 00 N N N N N
...(repeat for extents 04 through 16)....
NAME DAAXTK01 00 0 N N N N
NAME DAAXTK02 00 N N N N N
NAME DAAXTK03 00 N N N N N
...(repeat for extents 04 through 16)....
```

Repeat this process of changing from 0 to N for the same set of elements in the VCA\_VS parallel file further down in VCAGENIN.

- b. If you keep DAAXSZ01 and DAAXTK01 active (set to 0), you can still get the benefit of the Volume Problems Report (STGECA) produced by CA MICS StorageMate.
- c. Submit VCACGEN and ensure it completes without problems.

## 10.1.5 Alter the Active Timespans for a File

You can alter the active timespans and the status of the audit and history collection for a file by modifying the FILE statement for that file. You may also want to update the DBMODEL member of prefix.MICS.PARMS for an estimate of new space requirements.

See Section 4.2.4.2.1 of the System Modification Guide for more information about the FILE statement and see Section 7.3.4 of this guide for information about DBMODEL.

After the change has been made, you must run cccCGEN and CYCLEGEN in each unit that has the CA MICS Space Analyzer installed. If the status of the audit and history collection for a file has been changed, then the WEEKLY and MONTHLY JCL will have to be regenerated.

Example 1: Making VCA\_VS Inactive at DAYS and Above

Chapter 5 describes the VCA\_VS file in detail. After reading the sections in Chapter 5 that describe VCA\_VS and VCADAA, you will notice opportunities for consolidation.

If your MVS systems are 100% converted to ICF catalogs, then the VCADAA file can represent space allocated (DAASPACA, DAAALLSP, DAAKBYTA) information in a way perfectly suitable for accounting and capacity planning.

If you are willing to forego the historical data related to space used (DAASPACU, DAAUSESP, DAAKBYTU) kept in summary levels of the VCA\_VS file, then you could change the file statement for VCA\_VS to turn off the files at the DAYS level and higher.

a. Update sharedprefix.MICS.GENLIB(VCAGENIN):

Change the FILE statement for VCA\_VS from:

```
FILE _VSDAA 50 1 Y E N Y N Y N Y VSAM.....
```

to:

```
FILE _VSDAA 50 1 Y N N N N Y N N VSAM.....
```

For more information regarding the values in the FILE statement above, refer to Section 6.2.2 of the System Modification Guide (SMG).

b. Submit VCACGEN and ensure it runs correctly.

NOTE: By inactivating the VCA\_VS file at DAYS and above, some report paths in CA MICS StorageMate may be affected by the unavailability of data to produce a report. Check with all users of the data before performing this space-saving action.

Example 2: Making the VCADCS File Inactive

Chapter 5 describes how this file is already inactive at DETAIL. If you are not running DFSMS and using DATACLAS, or even if you are but are not interested in historical views of DATACLAS usage, then you can save space by making the VCADCS inactive at DAYS (this will eliminate it altogether).

Simply change the positional indicator at DAYS from E to N on the FILE DCS Statement and submit VCACGEN.

### 10.1.6 Add a New File

It may be necessary to add a new file to carry information that the CA MICS Space Analyzer does not support. The addition of a file requires the addition of FILE, FOPT, TYPE, and NAME statements to cccGENIN. To inform CA MICS about a possible change in the ccc SMF record number (even if you log records to the ccc DATA file and not to the SYS1.MANx structures) and to avoid possible data loss, it is also required to update the INPUTSOURCE value in sharedprefix.MICS.GENLIB(cccGENIN). Run an cccCGEN to activate the new definitions.

When updating cccGENIN, add the new file definition to the end of the member with a series of ascending sequence numbers that leave a significant gap at the end of the supplied statements. Following this technique satisfies IEBUPDTE and ensures that your modification can be easily adapted as future maintenance may update cccGENIN.

See Chapters 4 and 7 of the System Modification Guide for a complete discussion of this process.

## 10.1.7 Change the Number of Work Files for DAYnnn

NOTE: nnn is 090 for VCA component  
 or 095 for HSM component  
 ccc is VCA or HSM component

Follow the checklist below to change the number of work files used in the CA MICS Space Analyzer or its HSM component:

```
*****
*
* CHECKLIST 1 - CHANGE NUMBER OF WORK FILES
*
*****
```

- \_\_\_ 1. Update the WORK statement in prefix.MICS.PARMS(cccOPS) to specify the number of work files that are required.

```
+-----+
| Reference 7.3.1 for VCA Processing options (VCAOPS)|
|           7.3.2 for HSM Processing options (HSMOPS)|
+-----+
```

Example:

```
WORK n XXX pppp ssss
```

where: n (0-9) is the number of SAS WORK data sets  
 XXX is TRK or CYL  
 pppp is the primary allocation  
 ssss is the secondary allocation

examples: 1) WORK 6 TRK 800 100  
 2) WORK 4 CYL 100 20

In example 1, 6 SAS WORK data sets are specified, each with 800 tracks primary and 100 tracks secondary allocation.

In example 2, 4 SAS WORK data sets are specified, each with 100 cylinders primary and 20 cylinders secondary allocation.

Specify the minimum number of WORK data sets to meet your work space requirements. As a start, try specifying two or three work data sets.

- \_\_\_ 2. If this is the first time you are implementing multiple work files in CA MICS Space Analyzer or the HSM component, then continue with Step 2. If you are changing only the number currently in use or the space definitions, then proceed to Step 3 of this checklist.

Browse `sharedprefix.MICS.PROTOLIB(DYcccnnn)` checking for the presence of the `WORK` symbolic on the `EXEC` statement to determine if you have previously modified this product to increase the allocation of SAS `WORK` space.

- \_\_\_ 2a. If you find a `WORK` symbolic, divide the primary and secondary allocation values from the `WORK` symbolic by the number of work files that are specified earlier (value of `n` on the `WORK` statement that is coded in step 1).

Coding the resulting values yields the same aggregate space allocation that you have been using with a single SAS `WORK` file. To double your available `WORK` space, carry out the division, double the results and use the values in the previous `WORK` definition.

- \_\_\_ 2b. If you did not find a `WORK` symbolic in `PROTOLIB`, examine `prefix.MICS.PARMS(JCLDEF)` for each CA MICS unit containing the desired VCA or HSM component. Find the `WORKSPACE` keyword. The space allocation specified is used for a single SAS `WORK` file. Perform the same division as described in the previous paragraph to determine the quantity that yields the equivalent total allocation with multiple `WORK` files. Then adjust the values upward to meet your needs.

- \_\_\_ 3. Submit the job in `prefix.MICS.CNTL(cccPGEN)`.

- \_\_\_ 4. Edit `prefix.MICS.PARMS(JCLGENU)` so that it contains a single line that reads:

`DAILY`

- \_\_\_ 5. Submit the job in `prefix.MICS.CNTL(JCLGENU)`. Ensure that there are no error messages in `MICSLOG` or `SYSTSPRT`, that the `MICSLOG` contains the normal termination message, `BAS10999I`, and that the job completes with a condition code of zero.

\_\_\_ 6. The following operational jobs have changed:

#### DAILY

If your site has implemented the operational CA MICS processes in a scheduling product, the jobs may have to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

## 10.2 Standard User Exits

CA MICS defines a number of user exit points. Each of these points corresponds to a position in the CA MICS logic where user modification:

- o Is likely to occur, based on our experience.
- o Is unlikely to compromise CA MICS database and operational integrity.
- o Is relatively easy to document and understand.

Using the CA MICS exit facilities to modify CA MICS logic is the safest method of system modification. If you follow the material in this section, you can perform a substantial amount of system customization without causing any undue ramifications when you apply standard vendor maintenance to the product.

Before implementing an exit, determine whether you can accomplish the required modification by means of parameter modification or file tailoring. The CA MICS System Modification Guide provides full details of CA MICS exit processing in Section 4.3, User Exit Facilities.

The following sections identify the user exits available for CA MICS Space Analyzer processing, provide a general overview of the product's processing logic, and describe in detail each of the user exits.

- 1 - Available User Exits
- 2 - Additional Fields Available to Exits
- 3 - General Exits
- 4 - Output Processing Exits

## 10.2.1 Available User Exits

There are two types of user exits that you can use to modify CA MICS logic: general exits and product exits.

### GENERAL EXITS

All installed products use the general exits. They allow the insertion of user exit routines that extend CA MICS logic.

### PRODUCT EXITS

Product exits apply only to the product for which they are provided. They are classified as to their purpose and are comprised of input, output, parameter-related, and special exits.

The output exits are used when data is being written to the CA MICS database.

The parameter-related exits are also user exits, but are defined as part of the standard CA MICS installation process and are normally associated with other parameters or options. For example, the Account Code Routine is a user exit routine for entering the values for account codes that were defined in a separate but related installation.

Special exits are those that are classified as neither input, output, nor parameter-related. The calculation of a user-defined variable is an example of a type of special exit.

We have identified the user exits for the CA MICS Space Analyzer below:

#### General Exits

USRSEL - File Processing Selection  
USRIDL - Checkpoint File Inspection

#### Product Output Exits (VCA)

USRSDAA - Exit prior to Output of DETAIL.VCADAA File  
USRS\_VS - Exit prior to Output of DETAIL.VCA\_VS File  
USRVOA - Exit prior to Output of DETAIL.VCAVOA File  
USRSDCS - Exit prior to Output of DETAIL.VCADCS File  
USRSDIR - Exit prior to Output of DETAIL.HFSDIR File  
USRDDAA - Exit prior to Output of DAYS.VCADAA File  
USRD\_VS - Exit prior to Output of DAYS.VCA\_VS File

USRDDCS - Exit prior to Output of DAYS.VCADCS File  
USRDDIR - Exit prior to Output of DAYS.HFSDIR File  
USRUVOA - User Macro included in USRSVOA  
USRUDAA - User Macro included in USRSDAA  
USRU\_VS - User Macro included in USRS\_VS  
USRUDCS - User Macro included in USRSDCS  
USRUDIR - User Macro included in USRSDIR

#### Product Output Exits (HSM)

USRSMIG - Exit prior to Output of DETAIL.HSMMIG File  
USRSBAC - Exit prior to Output of DETAIL.HSBBAC File  
USRSVOL - Exit prior to Output of DETAIL.HSMVOL File  
USRSTAP - Exit prior to Output of DETAIL.HSMTAP File  
USRDMIG - Exit prior to Output of DAYS.HSMMIG File  
USRDBAC - Exit prior to Output of DAYS.HSBBAC File  
USRDVOL - Exit prior to Output of DAYS.HSMVOL File  
USRDTAP - Exit prior to Output of DAYS.HSMTAP File

#### Product Parameter-Related Exits

VCAACCT - DASD Account Code Assignment Exit  
HSMACCT - HSM Account Code Assignment Exit

#### Product Special Exits

USRUSEG - User segment input Exit for VCADAA file.  
USRHSEG - User segment input Exit for HSMMIG and HSBBAC files.

## 10.2.2 Additional Fields Available to Exits

The data elements available for use by the user exits vary according to the processing phase and the particular user exit. Special data elements that are used in the product update process but are not stored in the CA MICS database are described below. The individual user exit descriptions list which of these elements are available for the user exit routine.

### SMFRTYPE - SMF Record Type

This element contains the SMF record number that was written by the CA MICS Space Collector. This element is available to the USRSEL exit.

### DSNODE1-5 - Data Set Node Name Fields

The elements DSNODE1-5 contain the data set name qualifiers. DSNODE1 contains the highest level qualifier, DSNODE2 contains the next highest level qualifier, etc. These fields are created by parsing the data set name. These elements are not stored in the database. They are available for use by the user exits. Use these elements for assigning the DASD Space Analyzer account codes in member VCAACRT of sharedprefix.MICS.PARMS. The user exits USRSDAA and USRS\_VS also have access to these elements.

### DSNODEL - Last Node in Data Set Name

The element DSNODEL contains the lowest level qualifier in a data set name. This field is created by parsing the data set name. This element is not stored in the database. This field is available to the VCAACRT, USRSDAA and USRS\_VS exit routines.

### WRKRSRV1-2 - Reserved fields in Format 1 DSCB

Two elements WRKRSRV1 and WRKRSRV2 contain IBM reserved fields from the Format 1 DSCB in the VTOC. WRKRSRV1 contains the information read from offset 078-081 of the Format 1 DSCB in the VTOC, and WRKRSRV2 contains the information read from offset 103-104 of the Format 1 DSCB in the VTOC. These elements are not stored in the database. Certain DASD management products use these fields to store information such as the jobname that created the data set. These elements are available for use by the VCAACRT and USRSDAA exit routines.

FILEID - Current File Identifier for cccACRT

FILEID is set by calling module to the following values to give your exit code the opportunity to know for which file it has been called to assign account codes:

```
FILEID = 'VCADAA';    /* When called for VCADAA file */  
FILEID = 'VCA_VS';   /* When called for VCA_VS file */  
FILEID = 'HSMMIG';   /* When called for HSMMIG file */  
FILEID = 'HSBBAC';   /* When called for HSBBAC file */
```

### 10.2.3 General Exits

All products invoke general exits. They are presented here to identify the special uses available for the CA MICS Space Analyzer.

Each exit description includes the name and title, a description of its purpose, when it is invoked, and whether it has an interface to CA MICS Accounting and Chargeback. It also shows which data elements are available, and any special considerations.

```
+-----+  
| U S R S E L | - File Processing Selection Exit  
+-----+
```

DESCRIPTION: USRSEL allows access to each record processed by any product. You can use it to alter input data formats or to select or exclude specific records.

INVOCATION: This exit gains control after a detail input record has been read and before the record formats have been defined.

ACCOUNTING INTERFACE: No interface is provided.

USES: You may use this exit to delete records before processing or to alter the data, such as SMF SYSID.

ELEMENTS AVAILABLE:

ROUTINE - the name of the routine this exit invokes. To use this exit in the CA MICS Space Analyzer for the VCA component, test for:

```
ROUTINE = 'DYVCAFMT'  
SMFRTYPE - the record type of the input record  
ORGSYSID - original SYSID from the SMF record  
ENDTS - time when the record was written to SMF
```

To use this exit in the CA MICS Space Analyzer for the HSM component, test for:

```
ROUTINE = 'DYHSMFMT'  
SMFRTYPE - the record type of the input record  
ORGSYSID - original SYSID from the SMF record  
ENDTS - time when the record was written to SMF
```

CODING RESTRICTIONS: None

SPECIAL NOTES: Because all products use this exit, the source code is located in member #BASEXIT of the appropriate source libraries.

```
+-----+  
| U S R I H L | - Examine Input History Log  
+-----+
```

DESCRIPTION: USRIHL permits inspection of the DETAIL timespan checkpoint file immediately following processing of all input records in the format routine.

INVOCATION: This exit is referenced in code that passes the DETAIL timespan checkpoint file. This code reads the checkpoint file as an integrity check on that file, and in the process allows the user to inspect the file's contents.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is normally not of use to the CA MICS Space Analyzer.

ELEMENTS AVAILABLE: All elements in the ADMIHL file.

CODING RESTRICTIONS: None

SPECIAL NOTES: Because all products use this exit, the source code is located in member #BASEXIT of the appropriate source libraries.

## 10.2.4 Output Processing Exits

This section provides an in-depth description of the CA MICS Space Analyzer standard user exits that are invoked during the output phase of the daily update processing flow.

Each exit description includes the user exit name and title, a description of its purpose, and when it is invoked. It also shows what data elements are available and any special considerations.

Please note that the CA MICS Space Analyzer uses two forms of exits. They are Direct and Indirect exits. The Direct exits are used by the VCA component of the CA MICS Space Analyzer and the Indirect exits are used by the HSM component.

Exits must not issue the RETURN nor the DELETE statement since their invocation assumes the entire DATA step will be executed.

Exits can set SKIP\_REC = 1 to cause the current observation to be deleted.

This section is divided into the following subsections:

- 1 - Output Processing Exits for VCA
- 2 - Output Processing Exits for HSM

### 10.2.4.1 Output Processing Exits for VCA

The following exits use the direct form of exit invocation. For more information on how to do this, please refer to Section 4.3.1.1 of the System Modification Guide.

```
+-----+
| U S R S D A A | - Exit Prior to Output of DETAIL.VCADAA File
+-----+
```

DESCRIPTION: USRSDAA allows access to the data elements used to build the records in the DETAIL.VCADAA file. This file contains information from the VTOC that is collected by the CA MICS Space Collector.

INVOCATION: This exit gains control immediately prior to the output of a record to the DETAIL.VCADAA file.

USES: Use this exit to add or change data elements being written to the DETAIL.VCADAA file. You can also use this exit to set the value of the element DAACOST.

ELEMENTS AVAILABLE: All the elements that are turned on in VCAGENIN for the DAA file at the DETAIL timespan.

CODING RESTRICTIONS: None

SPECIAL NOTES: If CA MICS Accounting and Chargeback is active in the unit that contains the CA MICS Space Analyzer, DAACOST will be set by Accounting and Chargeback.

```
+-----+
| U S R U D A A | - Exit included in USRSDAA
+-----+
```

DESCRIPTION: The USRSDAA exit routines calls this user exit. Since it is called within the USRSDAA exit, it has access to the same information that the USRSDAA has.

INVOCATION: The USRSDAA exit routine invokes this user exit.

USES: You can code this exit in the #VCAEXIT member of prefix.MICS.USER.SOURCE instead of the USRSDAA exit. This will allow the exit logic to be unit dependent thus allowing greater flexibility.

ELEMENTS AVAILABLE: Same as the USRSDAA exit.

+-----+  
| U S R S \_ V S | - Exit Prior to Output of DETAIL.VCA\_VS File  
+-----+

DESCRIPTION: USRS\_VS allows access to the data elements used to build the records in the DETAIL.VCA\_VS file. This file contains VSAM data set information from catalogs read by the CA MICS Space Collector.

INVOCATION: This exit gains control immediately prior to the output of a record to the DETAIL.VCA\_VS file.

USES: You can use this exit to add or change data elements being written to the DETAIL.VCA\_VS file. You can also use it to set the value of the element DAACOST.

ELEMENTS AVAILABLE: All the elements that are turned on in VCAGENIN for the \_VS file at the DETAIL timespan.

SPECIAL NOTES: If CA MICS Accounting and Chargeback is active in the unit that contains the CA MICS Space Analyzer, DAACOST will be set by Accounting and Chargeback.

+-----+  
| U S R U \_ V S | - Exit included in USRS\_VS  
+-----+

DESCRIPTION: The USRS\_VS exit routine calls this user exit. Since it is called within the USRS\_VS exit, it has access to the same information that the USRS\_VS has.

INVOCATION: The USRS\_VS exit routine calls this user exit.

USES: This exit can be coded in the #VCAEXIT member of prefix.MICS.USER.SOURCE instead of the USRS\_VS exit. This will allow the exit logic to be unit dependent thus allowing greater flexibility.

ELEMENTS AVAILABLE: Same as the USRS\_VS exit.

+-----+  
| U S R S V O A | - Exit Prior to Output of DETAIL.VCAVOA File  
+-----+

DESCRIPTION: USRSVOA allows access to the data elements used to build the records in the DETAIL.VCAVOA file. This file contains Volume information from VTOCs and catalogs read by the CA MICS Space Collector.

INVOCATION: This exit gains control immediately prior to the output of a record to the DETAIL.VCAVOA file.

USES: Use this exit to add or change data elements being written to the DETAIL.VCAVOA file.

ELEMENTS AVAILABLE: All the elements that are turned on in VCAGENIN for the VOA file at the DETAIL timespan.

```
+-----+
| U S R U V O A | - Exit included in USRSVOA
+-----+
```

DESCRIPTION: The USRSVOA exit routine calls this user exit. Since it is called within the USRSVOA exit, it has access to the same information that the USRSVOA has.

INVOCATION: The USRSVOA exit routine calls this user exit.

USES: This exit can be coded in the #VCAEXIT member of prefix.MICS.USER.SOURCE instead of the USRSVOA exit. This will allow the exit logic to be unit dependent thus allowing greater flexibility.

ELEMENTS AVAILABLE: Same as the USRSVOA exit.

```
+-----+
| U S R S D I R | - Exit Prior to Output of DETAIL.HFSDIR File
+-----+
```

DESCRIPTION: USRSDIR allows access to the data elements used to build the records in the DETAIL.HFSDIR file. This file contains UNIX System Service's File System directory entries that are read by the CA MICS USS file systems Collector.

INVOCATION: This exit gains control immediately prior to the output of a record to the DETAIL.HFSDIR file.

USES: Use this exit to add or change data elements being written to the DETAIL.HFSDIR file.

ELEMENTS AVAILABLE: All the elements that are turned on in

VCAGENIN for the DIR file at the DETAIL timespan.

+-----+  
| U S R U D I R | - Exit included in USRSDIR  
+-----+

DESCRIPTION: The USRSDIR exit routine calls this user exit. Since it is called within the USRSDIR exit, it has access to the same information that the USRSDIR has.

INVOCATION: The USRSDIR exit routine calls this user exit.

USES: This exit can be coded in the #VCAEXIT member of prefix.MICS.USER.SOURCE instead of the USRSDIR exit. This will allow the exit logic to be unit dependent thus allowing greater flexibility.

ELEMENTS AVAILABLE: Same as the USRSDIR exit.

+-----+  
| U S R D D A A | - Exit prior to output of DAYS.VCADAA File  
+-----+

DESCRIPTION: USRDDAA allows access to the data elements used to build the records in the DAYS.VCADAA file. This file contains information from the VTOC that the CA MICS Space Collector collects.

INVOCATION: This exit gains control immediately prior to the output of a record to the DAYS.VCADAA file.

USES: Use this exit to add or change data elements being written to the DAYS.VCADAA file. Note that the observations being written have been summarized. You can use this exit to set DAACOST if it is not being set at the DETAIL level.

ELEMENTS AVAILABLE: All the elements that are turned on in VCAGENIN for the DAA file at the DAYS timespan.

CODING RESTRICTIONS: None

SPECIAL NOTES: If CA MICS Accounting and Chargeback is active in the unit that contains the CA MICS Space Analyzer, DAACOST will be set by Accounting and Chargeback.

+-----+

---

| U S R D \_ V S | - Exit prior to output of DAYS.VCA\_VS File  
+-----+

DESCRIPTION: USRD\_VS allows access to the data elements used to build the records in the DAYS.VCA\_VS file. This file contains VSAM data set information from catalogs read by the CA MICS Space Collector.

INVOCATION: This exit gains control immediately prior to the output of a record to the DAYS.VCA\_VS file.

USES: Use this exit to add or change data elements being written to the DAYS.VCA\_VS file. Note that the observations being written have been summarized. You can use this exit to set DAACOST if it is not being set at the DETAIL level.

ELEMENTS AVAILABLE: All the elements that are turned on in VCAGENIN for the \_VS file at the DAYS timespan.

SPECIAL NOTES: If CA MICS Accounting and Chargeback is active in the unit that contains the CA MICS Space Analyzer, DAACOST will be set by Accounting and Chargeback.

+-----+  
| U S R D V O A | - Exit prior to output of DAYS.VCAVOA File  
+-----+

DESCRIPTION: USRDVOA allows access to the data elements used to build the records in the DAYS.VCAVOA file. This file contains Volume information from VTOCs and catalogs read by the CA MICS Space Collector.

INVOCATION: This exit gains control immediately prior to the output of a record to the DETAIL.VCAVOA file.

USES: This exit can be used to add or change data elements being written to the DETAIL.VCAVOA file.

ELEMENTS AVAILABLE: All the elements that are turned on in VCAGENIN for the VOA file at the DAYS timespan.

+-----+  
| U S R D D I R | - Exit prior to output of DAYS.HFSDIR File  
+-----+

DESCRIPTION: USRDDIR allows access to the data elements used to build the records in the DAYS.HFSDIR file. This file

contains Directory entry information that is collected by USS File Systems Collector.

INVOCATION: This exit gains control immediately prior to the output of a record to the DAYS.HFSDIR file.

USES: Use this exit to add or change data elements being written to the DAYS.HFSDIR file.

ELEMENTS AVAILABLE: All the elements that are turned on in VCAGENIN for the DIR file at the DAYS timespan.

```
+-----+
| U S R D D C S | - Exit prior to output of DAYS.VCADCS File
+-----+
```

DESCRIPTION: USRDDCS allows access to the data elements used to build the records in the DAYS.VCADCS file. This file contains Data Class information for sites that have SMS implemented. This file is derived from the VCADAA file.

INVOCATION: This exit gains control immediately prior to the output of a record to the DAYS.VCADCS file.

USES: Use this exit to add or change data elements being written to the DAYS.VCADCS file.

ELEMENTS AVAILABLE: All the elements that are turned on in VCAGENIN for the DCS file at the DAYS timespan.

```
+-----+
| U S R U S E G | - User Segment Processing Exit
+-----+
```

DESCRIPTION: USRUSEG is called when a user segment is present in the records written by the CA MICS Space Collector. The Space Collector creates the user segments through the use of an exit called VCCUEXIT. This is provided for sites that have certain site specific information in the VTOC that is not read by the Space Collector. By coding the VCCUEXIT, this information can be written as part of the Space Collector records.

INVOCATION: This exit is called if a user segment exists else it is bypassed.

USES: Use this exit to read the user segment and store the

information into user defined elements.

ELEMENTS AVAILABLE: This should be determined by the installation.

SPECIAL NOTES: When the USRUSEG exit is called, the pointer in SAS is at the user segment. Therefore, it is not necessary to calculate the offset of the user segment from the start of the record. Use this logic to read the user segment which is as follows:

```
INPUT DAAXXXX PIB2.  
      DAAYYYY $CHAR3. @;
```

(DAAXXXX and DAAYYYY are user defined elements. In this example there are 2 fields in the user segment).

10.2.4.2 Output Processing Exits for HSM

Catalog Group ==> \*  
 Inquiry Name ==> \_\_\_\_\_

Line Cnds: I Insert D Delete S Select M Move E Exec-Time R Reset Parms

Cmd	Name	Title	Exec-Time	Catalog
-	-----	-----	-----	-----
-	RMFED1	MVS CPU and ASID Activity (.CSV)	*Specified*	SHARED
-	RMFED1	MVS CPU and ASID Activity (.CSV)	*Specified*	SHARED
-	RMFED5	MVS Top-4 CPU Busy Comparison (.CSV)		SHARED
-	RMFLPA	Post Processor CPU Activity Report		SHARED
-	CPUBSY	User Reporting CSV Extract		PRIVATE
***** BOTTOM OF DATA *****				

Scrolling RIGHT on an inquiry display shows the following:

SCROLL LEFT TO RETURN TO INQUIRY INFO

Line Cnds: I Insert D Delete S Select M Move E Exec-Time R Reset Parms

Inquiry

Cmd	Name	Typ	Prim	2nd	STORCLAS	DATACLAS	MGMTCLAS
-	-----	-----	-----	-----	-----	-----	-----
	RMFED1	CYL	25	5			
	RMFLPA						
	RMFCPU	CYL	15	5			
	PRTCPU	TRK	10	5			
***** Bottom of data *****							

The following exits use the indirect form of exit invocation.  
 For more information on how this is done, refer to Section 4.3.1.2 of the System Modification Guide.

```
+-----+
| U S R S V O L | - Exit prior to output of DETAIL.HSMVOL
+-----+
```

DESCRIPTION: USRSVOL allows access to the data elements used to build the records in the WORK.HSMVOL file. This file describes the characteristics of DASD volumes before and

after daily HSM space management, including percentage occupancy, target and trigger occupancy values, and excess eligible data occupancy.

INVOCATION: This exit gains control immediately prior to the output of a record to the WORK.HSMVOL file.

USES: This exit can be used to add or change data elements being written to the WORK.HSMVOL file.

ELEMENTS AVAILABLE: All the elements that are turned on in HSMGENIN for the VOL file at the DETAIL timespan.

CODING RESTRICTIONS: None

```
+-----+
| U S R D V O L | - Exit prior to output of DAYS.HSMVOL
+-----+
```

DESCRIPTION: USRDVOL allows access to the data elements used to build the records in the DAYS.HSMVOL file. This file describes the characteristics of DASD volumes before and after daily HSM space management, including percentage occupancy, target and trigger occupancy values, and excess eligible data occupancy.

INVOCATION: This exit gains control immediately prior to the output of a record to the DAYS.HSMVOL file.

USES: This exit can be used to add or change data elements being written to the DAYS.HSMVOL file.

ELEMENTS AVAILABLE: All the elements that are turned on in HSMGENIN for the VOL file at the DAYS timespan.

CODING RESTRICTIONS: None

```
+-----+
| U S R S B A C | - Exit prior to output of DETAIL.HSBBAC
+-----+
```

DESCRIPTION: USRSBAC allows access to the data elements used to build the records in the DETAIL.HSBBAC file. This file describes the SMS class attributes, number of backups, total space and current status for the incremental backups performed for data sets residing on primary DASD and under DFHSM incremental backup control.

INVOCATION: This exit gains control immediately prior to the output of a record to the DETAIL.HSBBAC file.

USES: This exit can be used to add or change data elements being written to the DETAIL.HSBBAC file.

ELEMENTS AVAILABLE: All the elements that are turned on in HSMGENIN for the BAC file at the DETAIL timespan.

+-----+  
| U S R D B A C | - Exit prior to output of DAYS.HSBBAC  
+-----+

DESCRIPTION: USRDBAC allows access to the data elements used to build the records in the DAYS.HSBBAC file. This file describes the SMS class attributes, number of backups, total space and current status for the incremental backups performed for data sets residing on primary DASD and under DFHSM incremental backup control.

INVOCATION: This exit gains control immediately prior to the output of a record to the DAYS.HSBBAC file.

USES: This exit can be used to add or change data elements being written to the DAYS.HSBBAC file.

ELEMENTS AVAILABLE: All the elements that are turned on in HSMGENIN for the BAC file at the DAYS timespan.

+-----+  
| U S R S M I G | - Exit prior to output of DETAIL.HSMMIG  
+-----+

DESCRIPTION: USRSMIG allows access to the data elements used to build the records in the DETAIL.HSMMIG00 file. This file describes the SMS class attributes, HSM level, space and current status for each data set in the installation that has been migrated from primary DASD.

INVOCATION: This exit gains control immediately prior to the output of a record to the DETAIL.HSMMIG00 file.

USES: This exit can be used to add or change data elements being written to the DETAIL.HSMMIG00 file.

ELEMENTS AVAILABLE: All the elements that are turned on in

HSMGENIN for the MIG file at the DETAIL timespan.

```
+-----+
| U S R D M I G | - Exit prior to output of DAYS.HSMMIG
+-----+
```

DESCRIPTION: USRDMIG allows access to the data elements used to build the records in the DAYS.HSMMIG00 file. This file describes the SMS class attributes, HSM level, space and current status for each data set in the installation that has been migrated from primary DASD.

INVOCATION: This exit gains control immediately prior to the output of a record to the DAYS.HSMMIG00 file.

USES: This exit can be used to add or change data elements being written to the DAYS.HSMMIG00 file.

ELEMENTS AVAILABLE: All the elements that are turned on in HSMGENIN for the MIG file at the DAYS timespan.

```
+-----+
| U S R S T A P | - Exit prior to output of DETAIL.HSMTAP
+-----+
```

DESCRIPTION: USRSTAP allows access to the data elements used to build the records in the DETAIL.HSMTAP file. This file describes the tapes under control of DFHSM for incremental backup, full volume backup, and HSM Level 2.

INVOCATION: This exit gains control immediately prior to the output of a record to the DETAIL.HSMTAP file.

USES: This exit can be used to add or change data elements being written to the DETAIL.HSMTAP file.

ELEMENTS AVAILABLE: All the elements that are turned on in HSMGENIN for the TAP file at the DETAIL timespan.

```
+-----+
| U S R D T A P | - Exit prior to output of DAYS.HSMTAP
+-----+
```

DESCRIPTION: USRDTAP allows access to the data elements used to build the records in the DAYS.HSMTAP file. This file describes the tapes under control of DFHSM for incremental

backup, full volume backup, and HSM Level 2.

INVOCATION: This exit gains control immediately prior to the output of a record to the DAYS.HSMTAP file.

USES: This exit can be used to add or change data elements being written to the DAYS.HSMTAP file.

ELEMENTS AVAILABLE: All the elements that are turned on in HSMGENIN for the TAP file at the DAYS timespan.

```
+-----+
| U S R H S E G | - User Segment Processing Exit
+-----+
```

DESCRIPTION: USRHSEG is called when a user segment is present in the VCCHSM data.

INVOCATION: This exit is called if a user segment exists else it is bypassed.

USES: This exit is used to read the user segment and store the information into user defined elements.

ELEMENTS AVAILABLE: This should be determined by the installation.

SPECIAL NOTES: When the USRHSEG exit is called, the pointer in SAS is at the user segment. Therefore, it is not necessary to calculate the offset of the user segment from the start of the record. The logic that should be used to read the user segment is as follows:

```
INPUT HSMUSRA1 $CHAR8. @;
```

(HSMUSRA1 is a user defined element)

Another feature that has been provided is the ability to carry these user defined elements in the data step by the use of the macro symbolic USRKEEP. Defining the user defined elements in the USRKEEP list ensures that the elements are available during the data step. For example, the user element HSMUSRA1 is defined to be in the keep list as follows.

```
%LET USRKEEP= HSMUSRA1;
```

## 10.3 File and Data Element Customization

A great amount of product tailoring can be performed by updating the product's parameters as described in Chapter 7 of this guide. That method addresses most of the data selection criteria. For example you can drop observations describing detailed extent information for VSAM data sets by simply specifying the EXTENTDETAIL option in VCAOPS.

Tailoring the database, as explained in Chapter 6 of the CA MICS System Modification Guide, is at the next level of detail. For example, deleting certain data elements of a file, adding others, creating a file in timespans that are not normally supported, and modifying the CA MICS archive process are not changes that can conveniently be made by the novice user.

Such changes to CA MICS are not to be taken lightly. The integrity and usability of the entire CA MICS database can be compromised by such changes. Therefore, thorough testing and verification is an absolute must.

Following the guidelines in Chapter 6 of the CA MICS System Modification Guide will help you tailor your CA MICS database. If, after you read that chapter, you still have a question on the feasibility or practicality of some intended change, please call the CA MICS Product Support Group for assistance.

## 10.4 Testing Modifications to the Analyzer

When you test modifications to the CA MICS Space Analyzer, you must do so in either a test unit or a test complex. If the modification requires a component generation (VCACGEN) to implement, use a test complex. Use a test unit if creating a test complex is not feasible or if the change does not require an VCACGEN to implement.

To test modifications, follow the steps below:

1. Make as few changes as possible. The simpler the change, the easier the comparison of "before" and "after" results.
2. Run the proper generation job(s) as required and as described in other sections.
3. Run update job(s) as required and as described in other sections.
4. Validate the data by comparing numbers of observations, relevant elements, and selected observations in the "before" and "after" runs.

## 10.5 A Case Study Using RACF and DMS/OS Data

One of the most frequent modifications made by clients with the CA MICS Space Analyzer is data capture from the security database and from the non-IBM fields of the format 1 DSCB. In this case study, we illustrate this extension with a series of examples of code that can be used at your site with appropriate modification and testing. Exits are always your responsibility; we provide a series of samples in this section to ease your implementation burden. If you have questions about these techniques, please call the CA MICS Product Support Group.

The following sections comprise this case study:

- 1 - Extract RACF and DMS/OS Data
- 2 - Modify VCAGENIN
- 3 - Modify #VCAEXIT - Define \_USRUSEG Macro
- 4 - Establish VCAACRT for User Data Elements
- 5 - Adapting Techniques for HSM Exits

## 10.5.1 Extract RACF and DMS/OS Data

The first step in this process involves extending the data collection function of VCC to extract data from the RACF database which can provide the departmental account code associated with a given data set (either non-VSAM or VSAM).

The mechanism for this is the VCCUEXIT, which is called by the VCCVTOC module for each format 1 DSCB found in the VTOC.

In the example that follows, we have also included the extraction of some data added to the format 1 DSCB by the Sterling Software product DMS/OS. The offsets into the format 1 DSCB are the default locations in Release 8.0 of DMS/OS according to our mutual clients. You must verify that your configuration of DMS/OS matches this; make the appropriate changes or delete the references pertaining to DMS/OS fields.

RACF examples are based on Release 1.8.1 of RACF. For a full discussion of VCCUEXIT, please refer to the VCC User Guide.

The aim of this exit is to extract the installation data (INSTDATA) that has been established for a RACF group profile. RACF and equivalent products set aside INSTDATA for use by the organization implementing the security environment. Naturally, our example here can only aim at giving you a view of one implementation and you will have to adapt this code to your own rules.



```

*****
LA    R3,VTCWUSVE      USER EXIT SAVE AREA      00500000
ST    R13,4(,R3)       CHAIN SAVE                00510000
ST    R3,8(,R13)       AREAS.                    00520000
LR    R13,R3           NEW SAVE AREA.            00530000
LA    R9,VTCWRECD      POINT TO SMF RECORD        00540000
USING SBASDSCT,R9      BASE SEGMENT BASE REG.    00550000
LH    R4,SBASRDW       - CURRENT RECORD LENGTH  00560000
N     R4,=X'FFFFFFFFC' - ENSURE FULLWORD ORIGIN  00570000
LA    R5,SUSRLEN       - LENGTH OF THE USER SEG.  00580000
STCM  R5,3,SBASUSRL   - USER SEGMENT LENGTH    00590000
AR    R5,R4            - NEW RECORD LENGTH      00600000
STH   R5,SBASRDW      - RESET SMF RECORD LENGTH  00610000
S     R4,=F'4'        - OFFSET OF USER SEGMENT  00620000
*                                     (SEGMENT OFFSETS DO NOT  00630000
*                                     INCLUDE THE FOUR BYTES  00640000
*                                     OF THE RDW FOR SAS).    00650000
STCM  R4,3,SBASUSRS   - USER SEGMENT OFFSET    00660000
LA    R4,4(R9,R4)     - FIND ADDRESS OF USER    00670000
*                                     SEGMENT                00680000
USING SUSRDSCT,R4     - USER SEG. BASE REG.    00690000
MVC   SBASUSR#,=AL2(1) - ONLY ONE USER SEGMENT  00700000
*                                     00710000
* BUILD USER SEGMENT FIELDS          00720000
*                                     00730000
*                                     00740000
*                                     00750000
*-----*                            00760000
* EXTRACT FIELDS ADDED TO FORMAT-1 DSCB BY DMS/OS * 00770000
*-----*                            00780000
MVC   SUSRCJOB(5),IECSDSL1+70 GET CREATING JOBNAME (1ST 5) 00790000
MVC   SUSRCJOB+5(3),IECSDSL1+45 (LAST 3 BYTES OF JOBNAME) 00800000
MVC   SUSRUJOB,IECSDSL1+62 GET LAST UPDATING JOB (ALL 8) 00810000
MVC   SUSRUDAT,IECSDSL1+48 DATE LAST UPDATED 00820000
*-----*                            00830000
* CALL RACF TO GET THE GENERIC PROFILE BASED ON COMPLETE DSNAME * 00840000
*-----*                            00850000
MVC   RACHECKW(RACHECKL),RACHECKX LIST FORM TO WORKAREA 00860000
RACROUTE REQUEST=AUTH, X00870000
      RELEASE=1.8.1, X00880000
      ENTITY=(DSIDSNAM,PRIVATE), X00890000
      WORKA=RACFWORK, X00900000
      RACFIND=NO, X00910000
      MF=(E,RACHECKW) 00920000
LTR   R15,R15         TEST RETURN CODE          00930000
BZ    MORERACF        0 - CONTINUE WITH MORE EXTRACTION 00940000
CH    R15,=H'8'      Q. GENERIC PROFILE FOUND ?      00950000
BNE   SETDEF         ..NO. GO SET DEFAULTS AND EXIT  00960000
MORERACF DS  0H      YES..CONTINUE                00970000
*-----*                            00980000

```

```

* CALL RACF TO GET INSTALLATION DATA BASED ON THE GENERIC PROFILE * 00990000
*-----* 01000000
      LA  R8,4(,R1)          -> GENERIC PROFILE 01010000
      MVC RACXTRTW(RACXTRTL),RACXTRTX LIST FORM TO WORKAREA 01020000
      RACROUTE REQUEST=EXTRACT, X01030000
          TYPE=EXTRACT, X01040000
          RELEASE=1.8.1, X01050000
          FIELDS=FLDLIST, X01060000
          DERIVE=YES, X01070000
          GENERIC=YES, X01080000
          WORKA=RACFWORK, X01090000
          ENTITY=(R8), X01100000
          MF=(E,RACXTRTW) 01110000
      LTR R15,R15 TEST RETURN CODE 01120000
      BNZ SETDEF GO SET DEFAULTS AND EXIT 01130000
      LH R5,4(,R1) GET OFFSET LENGTH 01140000
      AR R5,R1 -> INSTALLATION DATA 01150000
      CLI 0(R5),0 Q. TEST FOR NO INSTALLATION DATA 01160000
      BE EXIT ..NO INSTALLATION DATA A 01170000
      MVC ACCTNO,4(5) EXTRACT ACCOUNT CODE FROM RACF 01180000
      CLI ACCTNO,0 Q. SIMPLE TEST FOR VALIDITY 01190000
      BNE OK1 ..VALID...CONTINUE 01200000
      MVC ACCTNO,=CL8'*RACFERR' FOR LATER TESTING BY CA MICS 01210000
      B EXIT PROCEED TO EXIT 01220000
OK1 DS 0H 01230000
      CLC ACCTNO,=8CL1'00' Q. IS THERE AN ACCOUNT NUMBER 01240000
      BNE EXIT ..YES...EXIT 01250000
SETDEF DS 0H (ALSO) BRANCH ENTER IF NO PROFILE 01260000
EXIT DS 0H (ALSO) BRANCH ENTER IF NO PROFILE 01270000
      MVC ACCTNO,=8CL1'*' SET DEFAULT FOR TESTING BY CA MICS 01280000
      L R13,4(,R13) -> SAVEAREA PROVIDED AT ENTRY 01290000
      LM R14,R12,12(R13) RESTORE REGS AT ENTRY 01300000
      BR R14 RETURN TO VCCVTOC 01310000
      LTORG 01320000
*-----* 01330000
* CONSTANT AREA USED BY THIS CSECT * 01340000
*-----* 01350000
FLDLIST DC A(1) 01360000
        DC CL8'INSTDATA' 01370000
RACHECKX RACF1 01380000
RACHECKL EQU *-RACHECKX LIST LENGTH FOR MOVE TO WORKAREA 01390000
DUMMY DS 0H EXECUTE FORM OF MACRO OVERRIDES 01400000
RACXTRTX RACF2 01410000
RACXTRTL EQU *-RACXTRTX LIST LENGTH FOR MOVE TO WORKAREA 01420000
        SPACE 3 01430000
SUSRDSCT DSECT USER SEGMENT 01440000
SUSRCJOB DS CL8 CREATING JOBNAME 01450000
SUSRUJOB DS CL8 LAST UPDATING JOBNAME 01460000
SUSRUDAT DS XL3 DATE OF LAST UPDATE YY.DD.DD 01470000

```

```

SUSRACCT DS    CL8                ACCOUNT NUMBER FROM RACF INSTDATA  01480000
ACCTNO   EQU   SUSRACCT           ALIAS NAME FOR ACCOUNT NUMBER    01490000
SUSRLEN  EQU   *-SUSRDSCT         LENGTH OF USER SEGMENT           01500000
        SPACE 1                    01510000
RACHECKW RACF1                    SHELL FOR RACHECK MOVED HERE      01520000
RACXTRTW RACF2                    SHELL FOR RACXTRT MOVED HERE      01530000
        SPACE 2                    01540000
        DS    0D                    01550000
RACFWORK DS    CL512              WORKA ADDRESS FOR RACROUTE        01560000
        TITLE 'DSECTS NEEDED BY VCCUEXIT' 01570000
DSCB1    DSECT                    01580000
        IECSDSL1 1                  FORMAT 1 DSCB                    01590000
        PUSH PRINT                    01600000
        PRINT NOGEN                    01610000
        IEZDEB LIST=NO                DEB NEEDED BY VCCVTCWA MACRO      01620000
        POP PRINT                      01630000
        VCCVTCWA                      SAVEAREA/WORKAREA OF VCCVTOC TASK 01640000
        VCCCNTWA                      GLOBAL WORKAREA OF VCCNTRL        01650000
        VCCRCORD                      LAYOUT OF VCC RECORDS          01660000
        ICHSAFP                      01670000
        END VCCUEXIT                  01680000

```

Figure 10-1. Sample VCCUEXIT for RACF and DMS/OS

## 10.5.2 Modify VCAGENIN

The user elements stored in the VCC user segment need to be defined to the CA MICS file by updating the member VCAGENIN in sharedprefix.MICS.USER.GENLIB. Update VCAGENIN by defining the data elements as follows:

```

/* USER100 | 0791 | ADD LAST UPDATE DATE,JOBNAME,CREATE JOBNAME, RACF 00010050
NAME DAALUPDT 00 0 N N N N          DATE OF LAST UPDATE              00051100
TYPE R $8 . $8 . $8 .                00141100
NAME DAAUCJOB 00 0 N N N N          CREATING JOBNAME (DMS/OS)          00141200
NAME DAAUJJOB 00 0 N N N N          LAST UPDATE JOB (DMS/OS)          00141300

```

Note that the sequence numbers must be typed in as data in member VCAGENIN. Using these sequence numbers will ensure that the addition of these user fields will not collide with future CA MICS maintenance and enhancements. DAALUPDT is added immediately after DAAREFDT; it has the same format of DATE7.

### 10.5.3 Modify #VCAEXIT - Define \_USRUSEG Macro

The \_USRUSEG macro is defined as a stub in member #VCAEXIT in sharedprefix.MICS.SOURCE. To properly override this stub macro with your definition of the user segment, you need to create a local modification (for auditing purposes) which updates member #VCAEXIT in prefix.MICS.USER.SOURCE adding code after the %INCLUDE statement. Update #VCAEXIT by defining the SAS code for \_USRUSEG macro to read several user fields from the segment created by VCCUEXIT:

```
MACRO _USRUSEG
  INPUT  DAAUCJOB $CHAR8.
         DAAUUJOB $CHAR8.
         UPDTYEAR PIB1.
         UPDTDAY  PIB2.
        @;
  IF (0 < UPDTYEAR < 100) & (0 < UPDTDAY < 367) THEN
    DAALUPDT = DATEJUL(1000 * UPDTYEAR + UPDTDAY);

%                /* END OF OLD-STYLE MACRO */
```

You will also need to make sure that the WRKUSR01 element will be kept on the intermediate files WORK3.VCADAA00 and WORK.EXTDAA00.

A local modification that would work for this example could be stored as member LM0001 in sharedprefix.MICS.LOCALMOD.CNTL. In the example that follows, the first step saves a copy of the module prior to the application of LM0001. With this convention, the suffix of the local modification can point back to the module (or element) affected by the change.

```
//SM0001 EXEC PGM=IEBUPDTE
//I1 DD DSN=sharedprefix.MICS.SOURCE,DISP=OLD
//O1 DD DSN=sharedprefix.MICS.LOCALMOD.CNTL,DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  C I=((I1,R)),0=01
  S M=((DYVCAFMT,SM0001)) CREATE SAVED MODULE
//LM0001 EXEC PGM=IEBUPDTE
//SYSUT1 DD DSN=sharedprefix.MICS.SOURCE,DISP=OLD
//SYSUT2 DD DSN=sharedprefix.MICS.SOURCE,DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSIN DD DATA,DLM='++'
./ CHANGE NAME=DYVCAFMT
/* LM0001 | | ADD WRKUSR01 TO KEEP LISTS FOR PASSING RACF DATA */ 00010100
          WRKUSR01 /* FOR PHASE 6 DATA STEP USAGE */ 00050510
```

```

                WRKUSR01                /* FOR PHASE 6 DATA STEP USAGE */ 00054510
++
//

```

## 10.5.4 Establish VCAACRT for User Data Elements

The `_USRUSEG` macro is called during Phase 1 of the `DAY090` job step as the routine builds `WORK3.VCADAA00` and `WORK.EXTDAA00` from the raw VCC records. The modification to `VCAGENIN` described in Section 10.5.2 caused CA MICS to add the user data elements to the generated macros:

- o DAALen
- o DAAKEEP
- o DAAFMT

Therefore, now that the user data has had its length and format defined, has been extracted by means of the `INPUT` statements in the `_USRUSEG` macro, and has been kept as a variable on the `WORK3.VCADAA00` file, you must now finalize what you want to do with the accounting information obtained from the RACF installation data.

In the example below, the VCA Account Code variables are populated with data from RACF:

```

IF FILEID = 'VCADAA' THEN DO;
  VCAACT3 = 'ZFROMRACF';
  VCAACT2 = WRKUSR01;          /* ACCOUNT CODE FROM RACF */
  IF VCAACT2 = '*****' OR
     VCAACT2 = '*RACFERR' THEN DO;
    VCAACT3 = 'W' || DAAUCJOB;
    VCAACT2 = PUT(DAAUCJOB,$USRJOB.);
    IF VCAACT2 = '*****' THEN DO;
      /* SEE IF CREATED VIA TSO */
      VCAACT2 = PUT(DAAUCJOB,$USRUSER.);
      IF VCAACT2 = '*****' THEN DO;
        VCAACT3 = 'W*NOMATCH';
      END;
    END;
  END;
END;
VCAACT1 = PUT(VCAACT2,$USBILL.);
END;

```

The following notes may help to explain the above logic. Please bear in mind that this is only one possible implementation. Like all user exits, you are responsible for developing and testing.

- o There are 3 levels of account codes:
  - o VCAACT3 represents a USERID or an individual
  - o VCAACT2 represents the department to which the USERID is assigned
  - o VCAACT1 represents the division to which the department is assigned.
- o VCAACT3 is set to an arbitrary string (ZFROMRACF) to indicate that the department (account code) will be assigned from RACF data.
- o VCAACT2 is assigned from DAAARACF. This is the account code read earlier from the \_USRUSEG macro.
- o If the RACF administrator had not supplied the department account code as the installation data (INSTDATA) when setting up the GROUP PROFILE, then the previously discussed exits result in the VCAACT2 value being set to '\*\*\*\*\*' or '\*RACFERR'.
- o If that is the case, then:
  - o VCAACT3 is set to the value of the creating JOBNAME prefixed by an arbitrary character of 'W'.
  - o A PROC FORMAT table called \$USRJOB is searched for a match on the creating JOBNAME. Obviously, this presumes you validate JOBNAMEs and maintain this table to handle these RACF exceptions. The search returns the departmental account code or an arbitrary string of all '\*\*\*\*\*'.
  - o If not found in the \$USRJOB table, a search of another table, \$USRUSER, is done presuming the creating JOBNAME is a TSO USERID. If found, then the VCAACT2 variable is set to the departmental account code for that user. Note that this implies you have a job that can be run periodically or as new users are assigned which builds the PROC FORMAT which relates USERIDs to departments. If a USERID cannot be found, then we leave VCAACT2 set to all '\*' and set VCAACT3 to an arbitrary string of 'W\*NOMATCH'.
  - o Finally, VCAACT1 (the division) is determined by using the department account number (VCAACT2) as a search argument into a table (PROC FORMAT) called \$USBILL.

## 10.5.5 Adapting Techniques for HSM Exits

Sections in this chapter have discussed how RACF data can be obtained and integrated by a series of exits in the CA MICS DASD products:

- o VCCUEXIT to capture RACF (and DMS/OS) data
- o VCAGENIN modifications to define user fields
- o \_USRUSEG to read the user data
- o VCAACRT to translate user data into VCAACTx variables

Section 7.2 emphasizes the importance of synchronizing the setting of account code data between VCA and HSM. With the above techniques pertaining to VCA only, how can similar techniques be used for DFHSM data managed here by the HSM component of the product?

1. VCCUEXIT techniques can be cloned into a batch program which runs as a post-processing step to VCC. This program would simply:
  - a. OPEN the VCCHSM data file created by VCC
  - b. READ each record until EOF
  - c. For all type 'M' and type 'B' records, the RACF extraction logic shown in the VCCUEXIT could be used virtually without modification.
  - d. The account code would be placed in the HSM user segment and the main common segment of the record updated with the offset to the user segment.
  - e. At end-of-file, the data file is CLOSED.
2. VCAGENIN modification techniques could be similarly applied to HSMGENIN, but there is an alternative for HSM making this part easier. In  
shareprefix.MICS.USER.SOURCE(#HSMEXIT):

```
%LET USRKEEP =;
```

can be modified to the element or group of elements to be added to the KEEP lists generated for the HSMIG file and the HSBAC file. For example:

```
%LET USRKEEP = DEPTNO;
```

3. The USRHSEG macro plays the same role as the \_USRUSEG macro plays for VCA data. See Section 10.3 for details on USRHSEG.
4. Finally, the HSMACRT routine is used to for the same

purpose as for VCA when called for the HSMMIG file and the HSBBAC file. An example based on the example shown in Section 10.5.4 is shown here:

```
IF FILEID =: 'HS' THEN DO;
  HSMACT3 = 'ZFROMRACF';
  HSMACT2 = DEPTNO;          /* ACCOUNT CODE FROM RACF */
  END;
  HSMACT1 = PUT(HSMACT2,$USRBILL.);
END;
```

The only substantial difference between this and the example for VCA shown in Section 10.5.4 is the exception logic invoked if RACF administration had not established a department account number in the INSTDATA field of the RACF profile.

## 10.6 Change the Number of Work Files for DAY090

To change the number of work files used in the CA MICS Space Analyzer processing in Step DAY090, follow the checklist provided below for each unit.

See Section 7.3.1 of this guide for a detailed description of the statement syntax.

```
*****
*                                     *
*           CHANGE NUMBER OF WORK FILES           *
*                                     *
*****
```

- \_\_\_ 1. Update the WORK statement in prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier, to specify the number of work data sets required. Below is an example:

```
WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))
```

where:

n - is the number of WORK data sets.  
 STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.  
 SPACE - specifies how much disk space to provide for a new data set being allocated.  
 XXX - is TRK or CYL.  
 pppp - is the primary allocation.  
 ssss - is the secondary allocation.

You should specify the minimum number of WORK data sets to meet your work space requirements. As a start, try incrementing the number gradually beginning from the default.

- \_\_\_ 2. If this is the first time you are implementing multiple work files for this product, then continue with Step 2. If you are just changing the number currently in use, or simply the space definitions, then proceed to Step 3 of this checklist.

Browse sharedprefix.MICS.PROTOLIB(DYcccnnn) and sharedprefix.MICS.PROTOLIB(cccINCR), where (nnn) is the job step number and (ccc) is the product ID for this product, checking for the presence of the WORK

symbolic on the EXEC statement to determine if you have previously modified this product to increase the allocation of SAS WORK space.

- \_\_\_ 2a. If you find a WORK symbolic, simply divide the primary and secondary allocation values from the WORK symbolic by the number of work files specified above (value of n on the WORK statement coded in Step 1).

Coding the resulting values will yield the same aggregate space allocation you have been using with a single WORK file. To double your available WORK space, carry out the division, double the results and use the values in the WORK definition above.

- \_\_\_ 2b. If you did not find a WORK symbolic in PROTOLIB, examine prefix.MICS.PARMS(JCLDEF) for each CA MICS unit containing this product. Find the WORKSPACE keyword. The space allocation specified is used for a single SAS WORK file. Perform the same division as described in the previous paragraph to determine the quantity that will yield equivalent total allocation with multiple WORK files. Then adjust the values upward to meet your needs.

- \_\_\_ 3. Submit the job in prefix.MICS.CNTL(cccPGEN).

- \_\_\_ 4. If you specified RESTART YES in the product's cccOPS, you are done. Otherwise, you must do Steps 5, 6, and 7.

- \_\_\_ 5. Edit prefix.MICS.PARMS(JCLGENU) so that it contains a single line that reads:

DAILY

or, if incremental update is enabled for this product in this unit database, specify:

DAILY INCRccc

where ccc is the product ID.

- \_\_\_ 6. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.

\_\_\_ 7. The following operational job(s) have changed:

DAILY

INCRccc (if incremental update is enabled)

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

## 10.7 Change the Number of Work Files for DAY095

To change the number of work files used in the CA MICS Space Analyzer processing in Step DAY095, follow the checklist provided below for each unit.

See Section 7.3.2 of this guide for a detailed description of the statement syntax.

```
*****  
*                                                                 *  
*           CHANGE NUMBER OF WORK FILES                         *  
*                                                                 *  
*****
```

1. Update the WORK statement in prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier, to specify the number of work data sets required. Below is an example:

```
WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))
```

where:

n - is the number of WORK data sets.  
STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.  
SPACE - specifies how much disk space to provide for a new data set being allocated.  
XXX - is TRK or CYL.  
pppp - is the primary allocation.  
ssss - is the secondary allocation.

You should specify the minimum number of WORK data sets to meet your work space requirements. As a start, try incrementing the number gradually beginning from the default.

2. If this is the first time you are implementing multiple work files for this product, then continue with Step 2. If you are just changing the number currently in use, or simply the space definitions, then proceed to Step 3 of this checklist.

Browse sharedprefix.MICS.PROTOLIB(DYcccnnn) and sharedprefix.MICS.PROTOLIB(cccINCR), where (nnn) is the job step number and (ccc) is the product ID for this product, checking for the presence of the WORK

symbolic on the EXEC statement to determine if you have previously modified this product to increase the allocation of SAS WORK space.

- \_\_\_ 2a. If you find a WORK symbolic, simply divide the primary and secondary allocation values from the WORK symbolic by the number of work files specified above (value of n on the WORK statement coded in Step 1).

Coding the resulting values will yield the same aggregate space allocation you have been using with a single WORK file. To double your available WORK space, carry out the division, double the results and use the values in the WORK definition above.

- \_\_\_ 2b. If you did not find a WORK symbolic in PROTOLIB, examine prefix.MICS.PARMS(JCLDEF) for each CA MICS unit containing this product. Find the WORKSPACE keyword. The space allocation specified is used for a single SAS WORK file. Perform the same division as described in the previous paragraph to determine the quantity that will yield equivalent total allocation with multiple WORK files. Then adjust the values upward to meet your needs.

- \_\_\_ 3. Submit the job in prefix.MICS.CNTL(cccPGEN).

- \_\_\_ 4. If you specified RESTART YES in the product's cccOPS, you are done. Otherwise, you must do Steps 5, 6, and 7.

- \_\_\_ 5. Edit prefix.MICS.PARMS(JCLGENU) so that it contains a single line that reads:

DAILY

or, if incremental update is enabled for this product in this unit database, specify:

DAILY INCRccc

where ccc is the product ID.

- \_\_\_ 6. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.

\_\_\_ 7. The following operational job(s) have changed:

DAILY

INCRccc (if incremental update is enabled)

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

## 10.8 Enable Internal Step Restart

To enable the internal step restart in the CA MICS Space Analyzer, follow the checklist provided below:

See Section 7.3.1 and 7.3.2 of this guide for a detailed description of the statement syntax.

```
*****
*                                     *
*           ENABLE INTERNAL STEP RESTART           *
*                                     *
*****
```

- \_\_\_ 1. Edit prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier, and specify:

```
RESTART YES
```

For additional information on related topic, review the documentation for this product on WORK, RESTARTWORK, and RESTARTCKPT parameters to override default data set allocation parameters.

- \_\_\_ 2. Submit the job in prefix.MICS.CNTL(cccPGEN).
- \_\_\_ 3. Edit prefix.MICS.PARMS(JCLGENU) so that it contains a single line that reads:

```
DAILY
```

or, if incremental update is enabled for this product in this unit database, specify:

```
DAILY INCRccc
```

where ccc is the product ID.

- \_\_\_ 4. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.
- \_\_\_ 5. The following operational job(s) have changed:

```
DAILY
```

```
INCRccc (if incremental update is enabled)
```

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

## 10.9 Implement Incremental Update

To implement incremental update in the CA MICS Space Analyzer, follow the checklist provided below:

See Section 7.3.1 and 7.3.2 of this guide for a detailed description of the statement syntax.

```
*****
*                                     *
*           IMPLEMENT INCREMENTAL UPDATE           *
*                                     *
*****
```

- \_\_\_ 1. Edit prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier:
  - o Specify the following:
 

```
INCRUPDATE YES
```
  - o If you want to store the incremental update database files on tape between incremental updates, specify this:
 

```
INCRDB TAPE #gdgs
```
  - o If you want to allocate the incremental update database files during the first incremental update of the day and delete these data sets at the end of the DAILY job step, specify this:
 

```
INCRDB DYNAM
```
  - o If you specified INCRDB TAPE or INCRDB DYNAM, then you must also specify this:
 

```
INCRDETAIL data_set_allocation_parameters
INCRDAYS   data_set_allocation_parameters
```
  - o If you want the incremental update job for this product to get input measurement data from the output of the SPLITSMF job, specify this:
 

```
INCRSPLIT USE data_set_allocation_parameters
```
  - o For additional information on related topic, review the documentation for this product on INCRCKPT, INCRDETAIL, INCRDAYS, or INCRSPLIT

parameters to override default data set allocation parameters.

- \_\_\_ 2. Submit the job in prefix.MICS.CNTL(cccPGEN).
- \_\_\_ 3. Edit prefix.MICS.PARMS(JCLGENU) so that it contains two or more lines reading:  
  
DAILY  
INCRccc cccIUALC cccIUGDG
- \_\_\_ 4. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.
- \_\_\_ 5. Edit the job in prefix.MICS.CNTL(cccIUALC).
  - o Inspect and/or specify data set allocation parameters for the incremental update database and checkpoint files. If you specified INCRDB TAPE or INCRDB DYNAM, the cccIUALC job will only allocate the incremental update checkpoint data set.
  - o Submit the job. Ensure that there are no error messages in MICSLOG or SASLOG, and that the job completes with a condition code of zero.
- \_\_\_ 6. If you specified INCRDB TAPE, submit the job in prefix.MICS.CNTL(cccIUGDG) to define generation group indexes for the incremental update DETAIL and DAYS tape data sets. Examine SASLOG, MICSLOG, and SYSPRINT to verify that the generation group indexes were correctly defined.

Note: You may see error messages for the DLTX (or DELETE) statements. This is not a problem. cccIUGDG deletes each index prior to defining it, and an error message is issued if the index does not yet exist (e.g., if this is the first time you ran the cccIUGDG job).

- \_\_\_ 7. The following operational job(s) have changed:

DAILY INCRccc

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have

to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

- \_\_\_ 8. Implement operational procedures for gathering input measurement data and executing incremental updates (INCRccc) during the day.

You may also need to modify operational procedures for the DAILY job to ensure that processing is limited to input measurement data that has not been input to one of the day's incremental update executions.



# Appendix A: Messages

---

This section contains the following topics:

[A.1 VCA Component Messages](#) (see page 391)

[A.2 HSM Component Messages](#) (see page 421)

## A.1 VCA Component Messages

This appendix lists all messages generated by the VCA component of the CA MICS Space Analyzer. Some messages are generated during the processing of the control statements, while others are caused by various conditions in the data found during the processing. The messages are listed in ascending numerical sequence and include the full text of the message, the type, the reason for the message, appropriate user action, and applicable references to documentation.

The following type codes are used to categorize the messages:

Information	Designates a note that documents a CA MICS option or potentially important feature in the data.
Warning	Designates a condition in either the data or the control statements that does not affect the CA MICS Space Analyzer's operation, but that may lead to unexpected results.
Error	Designates that a problem has been encountered with a control statement that will prevent a successful run of the product. Execution is stopped after all control statements are processed.

The message text often includes references to information that is contained in a control statement or values of permanent or temporary variables created during CA MICS processing. In the description below, the message text contains a word beginning with a percent sign (%), which indicates that a value will be substituted into the text at execution time. For example, message VCA00016 reads as follows:

```
TEXT: >>>> SYSID NOT 1 TO 4 CHARACTERS IN LENGTH.  
      SYSID=%TOKEN2.
```

A possible version of the printed message in the MICSLOG might be:

```
TEXT: >>>> SYSID NOT 1 TO 4 CHARACTERS IN LENGTH.  
      SYSID=SYS008 .
```

```
+-----+  
| V C A 0 0 0 0 1 |  
+-----+
```

TEXT: VCA PARAMETER GENERATION STARTED.

TYPE: Information

REASON: The beginning of the VCAPGEN job is noted.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 0 0 2 |  
+-----+
```

TEXT: VCA PARAMETER GENERATION COMPLETED SUCCESSFULLY.

TYPE: Information

REASON: The end of the VCAPGEN job is noted.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 0 1 0 |  
+-----+
```

TEXT: ---- VCAOPS MEMBER:

TYPE: Information

REASON: The content of the prefix.MICS.PARMS member  
called VCAOPS follows this message line.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 0 1 1 |  
+-----+
```

TEXT: %LINE

TYPE: Information

REASON: This is a general purpose message used by both VCAPGEN processing and by the DAILY job (step DAY090). The value of the variable LINE is displayed on the MICSLOG output data set for informational purposes.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 0 1 5 |  
+-----+
```

TEXT: >>>> KEYWORD NOT RECOGNIZED KEYWORD=%TOKEN1.

TYPE: Error

REASON: The displayed keyword is not defined for VCAPGEN.

ACTION: Correct the keyword and resubmit the job.

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 0 1 6 |  
+-----+
```

TEXT: >>>> SYSID NOT 1 TO 4 CHARACTERS IN LENGTH.  
SYSID=%TOKEN2.

TYPE: Error

REASON: SYSID cannot exceed 4 characters.

ACTION: Correct the OPTIONS statement value for SYSID or consider using '\*' to represent all SYSID values.

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 0 1 7 |  
+-----+
```

TEXT: >>>> INTERVAL TIME VALUE MUST BE 1-999 HOURS.  
VALUE=%TOKEN3.

TYPE: Error

REASON: The default duration value is not within the proper range. This value should be based on how frequently the CA MICS Space Collector job (VCC scan) is run, creating input for this database unit. For example, if the VCC scan is run once per day, this value should be set to 24 to represent 24 hours.

On the other hand, if VCC scan is run hourly, then this parameter should be set to 1.

ACTION: Correct the OPTIONS statement value for the default duration and resubmit.

REFERENCES: Section 7.3.1

```
+-----+
| V C A 0 0 0 2 0 |
+-----+
```

TEXT: ---- END OF VCAOPS MEMBER.

TYPE: Information

REASON: The content of the prefix.MICS.PARMS member has been displayed above this message line.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+
| V C A 0 0 0 2 4 |
+-----+
```

TEXT: OPTIONS STATEMENT MISSING. ASSUMING DURATION OF 24 HOURS AND ACCEPTANCE OF INPUT FROM ANY SYSID.

TYPE: Information

REASON: Options statement in VCAOPS is missing. A default DURATION of 24 hours is being used and data from all SYSIDS will be processed.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 0 2 5 |  
+-----+
```

TEXT: BCSREQUIRED STATEMENT MISSING. ASSUMING NO  
REQUIREMENT FOR BCS (ONLY VVDS) FOR VSAM.

TYPE: Information

REASON: BCSREQUIRED statement in VCAOPS is missing.  
Default being used is BCSREQUIRED NO.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 0 2 6 |  
+-----+
```

TEXT: EXTENTDETAIL STATEMENT MISSING. ASSUMING NO  
RETENTION FOR DETAIL EXTENT DATA BEYOND 16  
EXTENTS.

TYPE: Information

REASON: EXTENTDETAIL statement in VCAOPS is missing.  
Default being used is EXTENTDETAIL NO.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 0 2 7 |  
+-----+
```

TEXT: ACCOUNTING STATEMENT MISSING. ASSUMING  
COMPUTATION OF DASD COST AT -DAYS- TIMESPAN.

TYPE: Information

REASON: ACCOUNTING statement in VCAOPS is missing.  
Default being used is ACCOUNTING DAYS.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+
| V C A 0 0 0 2 8 |
+-----+
```

TEXT: INVALID PARAMETER FOR %TOKEN1 KEYWORD. SPECIFY  
%CHOICE1 OR %CHOICE2.

TYPE: Error

REASON: The syntax for this statement requires the  
choice between two parameter alternatives.

ACTION: Correct the VCAOPS member and resubmit the job.

REFERENCES: Section 7.3.1

```
+-----+
| V C A 0 0 0 3 0 |
+-----+
```

TEXT: >>>> ERROR(S) ENCOUNTERED IN VCAOPS PARM  
DEFINITIONS.

>>>> \$VCAMSTR GENERATION ABORTED.

>>>> CORRECT ERROR(S) AND EXECUTE AGAIN.

TYPE: Error

REASON: VCAPGEN processing has encountered a syntax  
error.

ACTION: Correct the errors as indicated by prior messages  
and resubmit VCAPGEN.

REFERENCES: Section 7.3.1

```
+-----+
| V C A 0 0 0 7 5 |
+-----+
```

TEXT: VCACGEN option default taken  
Option: PDSEUSED/NOPDSEUS Default: NOPDSEUS

TYPE: Information

REASON: The OPTION statement in  
sharedprefix.MICS.GENLIB(VCAGENIN) did not  
specify a value, so the default value was used.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 1 0 0 |  
+-----+
```

TEXT: ----- BEGIN PHASE 1 -----

TYPE: Information

REASON: This message is displayed during the building of the first observation in the first DATA step of the DAY090 job step.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 1 0 1 |  
+-----+
```

TEXT: AT LEAST ONE RECORD WITH SYSID NOT DEFINED IN VCAOPS (%ORGSYSID). THE DEFAULT DURATION WILL BE USED.

TYPE: Warning

REASON: A SYSID that is not defined in VCAOPS has been encountered in the input data being processed. Default DURATION will be assigned to this SYSID.

ACTION: If you want to specify a DURATION for this SYSID, update VCAOPS and run VCAPGEN.

REFERENCES: Section 7.3.1

```
+-----+  
| V C A 0 0 1 0 2 |  
+-----+
```

TEXT: AT LEAST ONE RECORD READ WITH BAD VCC RECORD ID. (%WRKID). SUCH RECORDS ARE DELETED.

TYPE: Warning

REASON: A record was found with a record ID that is

invalid. The record ID must be 'VCC'. The records are deleted and the option deleted counter is incremented.

ACTION: Check to see if invalid data is being input to the DAY090 step.

REFERENCES: None

```
+-----+  
| V C A 0 0 1 0 3 |  
+-----+
```

TEXT: VCC VERSIONS PRIOR TO 101 ARE NO LONGER SUPPORTED

TYPE: Error

REASON: The input data being processed is from a version of the Space Collector that is not supported.

ACTION: Apply required VCC maintenance from the PSP tape. For assistance, contact Technical Support at <http://ca.com/support>.

REFERENCES: None

```
+-----+  
| V C A 0 0 1 0 4 |  
+-----+
```

TEXT: FIRST INPUT TIMESTAMP THIS RUN=%ENDTS FROM SYSTEM %ORGSYSID.

TYPE: Information

REASON: This message is issued for the first record read by VCA. It gives the ENDTS and the SYSID.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 1 1 0 |  
+-----+
```

TEXT: >>>> VOL=%VOLSER DSN=%DSNAME NUMBER OF DATA SET EXTENTS IS INVALID. EXTNO=%DAAEXTNO

TYPE: Error

REASON: The extent information for the data set being processed is invalid.

ACTION: Use ISPF to verify that the DSNAME is valid.  
Consider using a function to consolidate the data set into one extent.

REFERENCES: None

```
+-----+  
| V C A 0 0 1 1 1 |  
+-----+
```

TEXT: >>>> SYSID=%SYSID VOL=%VOLSER DSN=%DSNAME  
----- DS1LSTAR VALUE FROM VTOC IS INVALID.

TYPE: Warning

REASON: The DS1LSTAR value read from the VTOC is invalid.  
Using the value would result in invalid space used value. Space used is set equal to space allocated.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 1 1 2 |  
+-----+
```

TEXT: >>>> SYSID=%SYSID VOL=%VOLSER DSN=%DSNAME  
----- SPACE USED IS GREATER THAN SPACE ALLOCATED.

TYPE: Warning

REASON: The DS1LSTAR value read from the VTOC is invalid.  
Because space used is greater than space allocated, space used is set to be equal to space allocated.

ACTION: Data sets with an invalid DS1LSTAR should be investigated and repaired if possible with the appropriate utility function (COPY, COMPRESS, EDIT(SAVE), etc).

REFERENCES: Section 4.3, Exception Report EXC09019.

```
+-----+
| V C A 0 0 1 1 3 |
+-----+
```

TEXT: >>>> SYSID=%SYSID VOL=%VOLSER DSN=%DSNAME  
----- KEY RANGE QUALIFIER BEYOND END OF RECORD -  
SET TO BLANKS.

TYPE: Warning

REASON: The input record from VCC has an internal pointer  
problem so that the key range qualifier cannot be  
read from that record.

ACTION: This problem likely originates with the VSAM file  
and its catalog data. Examine it and possibly  
correct problems by deleting and redefining the  
data set.

REFERENCES: None

```
+-----+
| V C A 0 0 1 1 4 |
+-----+
```

TEXT: >>>> SYSID=%SYSID VOL=%VOLSER DSN=%DSNAME  
----- EXTENT INFO BEYOND END OF RECORD - INFO LOST.

TYPE: Warning

REASON: The input record from VCC has an internal pointer  
problem so that VSAM extent information cannot be  
read from that record.

ACTION: This problem likely originates with the VSAM file  
and its catalog data. Examine it and possibly  
correct problems by deleting and redefining the  
data set.

REFERENCES: None

```
+-----+
| V C A 0 0 1 1 5 |
+-----+
```

TEXT: >>>> SYSID=%SYSID VOL=%VOLSER DSN=%DSNAME  
----- INVALID LOW/HIGH KEY LENGTH FOUND, SETTING TO  
BLANKS.

TYPE: Warning

REASON: The input record from VCC has an internal pointer problem so that either the VSAM low key or high key cannot be read from that record.

ACTION: This problem likely originates with the VSAM file and its catalog data. Examine it and possibly correct problems by deleting and redefining the data set.

REFERENCES: None

```
+-----+  
| V C A 0 0 2 0 0 |  
+-----+
```

TEXT: ----- BEGIN PHASE 2 -----

TYPE: Information

REASON: This message is displayed during the start of phase 2 of the DAY090 job step.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 3 0 0 |  
+-----+
```

TEXT: ----- BEGIN PHASE 3 -----

TYPE: Information

REASON: This message is displayed during the start of phase 3 of the DAY090 job step.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 3 0 1 |  
+-----+
```

TEXT: CATALOG RESTART REPORT

THE FOLLOWING VOLUMES WERE NOT SUCCESSFULLY  
 PROCESSED BY VCC. ALL CATALOGED DATA SETS  
 COLLECTED FROM THESE VOLUMES WILL BE EXCLUDED FROM  
 THE VCA\_VS FILE.

TYPE	CLUSTER NAME	DATA SET NAME
----	-----	-----

TYPE: Warning

REASON: This report will be printed as part of the MICSLLOG if the Space Collector fails to completely process a volume. It documents that the VCC records collected from your catalogs describing the VSAM data sets on these volumes have been received and ignored by the Space Analyzer. This is an informational report and shows that the normal error detection and restart process is working correctly.

ACTION: The Space Collector's output should be examined to determine the problem.

REFERENCES: None

```
+-----+
| V C A 0 0 3 0 2 |
+-----+
```

TEXT: >>>> VOLUME=%VOLSER - SYSID=%SYSID %ENDTS -  
 VOL. NOT SUCCESSFULLY PROCESSED

TYPE: Warning

REASON: This message is issued for every volume that is not successfully processed by the Space collector. The data set information collected from the indicated volume will not be written to the VCA\_VS file.

ACTION: The Space Collector's output should be examined to determine the problem.

REFERENCES: None

```
+-----+
| V C A 0 0 3 0 8 |
+-----+
```

TEXT: >>>> %dNNNNN DATA SETS IGNORED FROM %WRKMSG2  
VOLUMES.

TYPE: Warning

REASON: This message gives the total number of data sets  
whose information is not written to the VCA\_VS  
file.

ACTION: See the VCA00302 message that precedes this  
message for the action that must be taken.

REFERENCES: None

```
+-----+  
| V C A 0 0 4 0 0 |  
+-----+
```

TEXT: ----- BEGIN PHASE 4 -----

TYPE: Information

REASON: This message is displayed during the start of  
phase 4 of the DAY090 job step.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 4 0 1 |  
+-----+
```

TEXT: ATTRIBUTES AND STATISTICS EXCEPTION REPORT

THE PRIMARY VOLUME FOR THE FOLLOWING VSAM DATA SETS  
WAS NOT PROCESSED BY VCC. SINCE THE ATTRIBUTES AND  
STATISTICS ELEMENTS ARE ONLY PRESENT IN THE PRIMARY  
VOLUME VVDS, MANY ELEMENTS WILL BE ZERO IN THE  
VCA\_VS FILE.

TYPE	VOLUME	CLUSTER NAME	DATA SET NAME
----	-----	-----	-----

TYPE: Warning

REASON: VSAM data sets allocated on multiple volumes or

split into multiple key ranges have important data recorded only in the first or primary volume's VVDS. This information affects 37 data elements that must be copied to the other VCA\_VS observations collected for this data set. This report will be printed as part of the MICSL0G if the primary volume's AMDSB is not present (the primary volume was excluded from VCC or VCC volume processing failed).

**ACTION:** The Space Collector output may need to be examined to determine if some volumes are not being scanned. If some volumes are being specifically excluded from the Space Collector Scan job and those volumes are the primary volumes for the VSAM data sets indicated following this message, then no action may be necessary. However, the impact of many elements in the VCA\_VS file for these data sets being zero must be considered.

**REFERENCES:** None

```
+-----+
| V C A 0 0 4 0 2 |
+-----+
```

**TEXT:** >>>> SYSID=%SYSID %ENDTS - DATA SETS CONTAIN  
INCOMPLETE DATA IN VCA\_VS

**TYPE:** Warning

**REASON:** This message is issued after VCA00401.

**ACTION:** See VCA00401W and VCA00403W.

**REFERENCES:** None

```
+-----+
| V C A 0 0 4 0 3 |
+-----+
```

**TEXT:** %DAVCOMP %VOLSER %DAACNAME %DSNAME

**TYPE:** Warning

**REASON:** The data sets that have incomplete information due to reasons described for the VCA00401W message are indicated.

ACTION: See the VCA00401W message.

REFERENCES: None

```
+-----+  
| V C A 0 0 4 1 0 |  
+-----+
```

TEXT: %SYSID %DAAVCOMP %VOLSER %DAACNAME %DSNAME  
>>>> SPACE USED BY THE ABOVE DATA SET IS  
GREATER THAN THE SPACE ALLOCATED.

TYPE: Warning

REASON: The space used is found to be greater than the  
space allocated. Possible corrupted values in the  
catalog. The value of space used is set to be  
equal to space allocated.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 4 1 1 |  
+-----+
```

TEXT: VSAM EXCEPTION REPORT

THE FOLLOWING VOLUMES CONTAIN VSAM OBJECTS THAT  
DO NOT HAVE VCC RECORDS FROM BOTH  
A BASE CATALOG STRUCTURE (BCS) AND A VSAM  
VOLUME DATA SET (VVDS) OR THAT ARE  
DUPLICATED IN AN OLD-STYLE VSAM CATALOG, BCS,  
AND/OR VVDS.

TYPE: Information

If you have turned the VVDS/BCS merge processing  
option back on as described earlier (VCAXBCS macro  
symbol changed from NO to YES), then all VSAM data  
sets must have a VVDS record and a matching BCS  
record. When a VSAM data set has only a VVDS  
record, it is listed in this report and excluded  
from the VCA\_VS file.

Because the BCS record contains only seven data  
elements, this report will be produced regardless  
of the VCAXBCS value when a VSAM data set has a

BCS record but does not have a matching VVDS record. The BCS record describing the VSAM data set will be listed but will not be added to the VCA\_VS file. You must research the Space Collector run to determine why the VVDS was not processed.

This report is also produced if a VSAM data set has records collected from two or more old-style VSAM catalogs, ICF Base Catalogs, or VVDS catalogs. All duplicate defined VSAM data sets are listed and are accompanied by VCA00413W, VCA00417I, and depending upon the type of error encountered, messages VCA00414W, VCA00415W, or VCA00416W.

**ACTION:** See VCA00417I for the Cluster, data set name, and catalog name. Messages VCA00414W, VCA00415W, and VCA00416W indicate the specific reasons for the data set information not being written to the VCA\_VS file.

**REFERENCES:** None

```
+-----+
| V C A 0 0 4 1 2 |
+-----+
```

TEXT:	TYPE (CAT)	CLUSTER NAME DATA SET NAME	REASON CATALOG NAME
----	-----	-----	-----

**TYPE:** Information

**REASON:** This is a header message for VCA00414I, VCA00415I, VCA00416I, and VCA00417I messages.

**ACTION:** None

**REFERENCES:** None

```
+-----+
| V C A 0 0 4 1 3 |
+-----+
```

TEXT:	TYPE (CAT)	CLUSTER NAME DATA SET NAME	REASON CATALOG NAME
----	-----	-----	-----

**TYPE:** Warning

**REASON:** This message is issued for every volume that has

data sets that were not written to the VCA\_VS file.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 4 1 4 |  
+-----+
```

TEXT: %DAACNAME DUPLICATE CATALOG ENTRIES

TYPE: Warning

REASON: Duplicate catalog entries have been detected for the data set indicated by VCA00417I message. DAACNAME is the Cluster name. The data set information is not written to the VCA\_VS file.

ACTION: Contact the DASD management group at your site to resolve the problem.

REFERENCES: None

```
+-----+  
| V C A 0 0 4 1 5 |  
+-----+
```

TEXT: %DAACNAME NO MATCHING VVDS RECORD

TYPE: Warning

REASON: An entry for a VSAM data set was found in the BCS catalog but no matching VVDS record was found.

ACTION: The data set information is not written to the VCA\_VS file. This could mean that there is an entry in the catalog regarding this VSAM data set although the data set no longer exists or the volume on which the data set resides may be excluded from the Space Collector scan job. If the problem is the first case, contact the DASD management group at your site to correct this situation.

REFERENCES: None

```
+-----+
```

```
| V C A 0 0 4 1 6 |
+-----+
```

TEXT: %DAACNAME NO MATCHING BCS RECORD

TYPE: Warning

REASON: This message is issued when there is an entry for a VSAM data set in the VVDS but not in any BCS. BCSREQUIRED YES option has been specified in VCAOPS indicating that an entry must be found for a VSAM data set in the BCS catalog as well as the VVDS so that the data set information can be written to the VCA\_VS file. The observation is not written to the VCA\_VS file.

ACTION: This situation could occur if the BCS catalog is excluded from the Space Collector scan job. If this is not the reason, then you may need to contact the DASD management group at your site to report this situation.

REFERENCES: None

```
+-----+
| V C A 0 0 4 1 7 |
+-----+
```

TEXT: %DAAVCOMP (%WRKTYPE) %DSNAME CATALOG = %WRKCATNM

TYPE: Information

REASON: This message is issued after VCA00414W, VCA00415W, or VCA00416W messages. It indicates whether the data set is a sequence set data set, the data set name, and the catalog name and whether the entry was found in a BCS or VVDS (B or V) catalog.

ACTION: None

REFERENCES: None

```
+-----+
| V C A 0 0 4 1 8 |
+-----+
```

TEXT: >>>> %NNNNN VSAM DATA SETS DELETED FROM  
%VVVVV VOLUMES.

TYPE: Warning

REASON: This message gives the total count of all the VSAM data sets (NNNNN) from (VVVVV) whose information is not written to the VCA\_VS file.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 4 1 9 |  
+-----+
```

TEXT: >>>> %WRKMSG1 VSAM DATA SETS DELETED FROM VCA\_VS VIA USRS\_VS EXIT.

TYPE: Information

REASON: This message is issued if data sets were deleted by means of the USRS\_VS exit. This is a user-defined exit.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 5 0 0 |  
+-----+
```

TEXT: ----- BEGIN PHASE 5 -----

TYPE: Information

REASON: This message is displayed during the start of phase 5 of the DAY090 job step.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 5 0 1 |  
+-----+
```

TEXT: VSAM/VOLUME EXCEPTION REPORT THE FOLLOWING VOLUME(S) WERE NOT PROCESSED BY VCC. ALL VSAM DATA SETS COLLECTED ARE ADDED TO THE VCA\_VS FILE BUT ARE

NOT SUMMARIZED IN THE VCAVOA FILE.

TYPE	CLUSTER NAME	DATA SET NAME
-----	-----	-----

TYPE: Information

REASON: This report title message is issued when volume level observations were not found for certain volumes. The volume level information for the volume indicated in VCA00502W message is not written to the VCAVOA file.

ACTION: See the VCA00502W message for action that may need to be taken.

REFERENCES: None

```
+-----+
| V C A 0 0 5 0 2 |
+-----+
```

TEXT: >>>> VOLUME=%VOLSER - SYSID=%SYSID %WRKENDTS -  
VSAM DATA SETS HAVE NO VCAVOA OBSERVATION

TYPE: Warning

REASON: Volume level information has not been found for the volume indicated in the message. The VCAVOA file will not contain any information regarding this volume.

ACTION: The Space Collector scan job output needs to be examined to identify why the volume level information records were not written.

REFERENCES: None

```
+-----+
| V C A 0 0 5 0 3 |
+-----+
```

TEXT: %DAVCOMP %DAACNAME %DSNAME

TYPE: Information

REASON: This message is issued after VCA00502W. The data sets indicated by this message would have contributed to space calculations in the VOA file

for the volume indicated in the VCA00502W message.  
The information regarding these data sets is  
present in the VCA\_VS file.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 5 0 9 |  
+-----+
```

TEXT: >>>> %WRKMSG1 VOLUME RECORDS DELETED FROM VCAVOA  
VIA USRSVOA EXIT.

TYPE: Information

REASON: This message is issued if data sets were deleted  
by means of the USRSVOA exit. This is a user-  
defined exit.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 0 |  
+-----+
```

TEXT: ----- BEGIN PHASE 6 -----

TYPE: Information

REASON: This message is displayed during the start of  
phase 6 of the DAY090 job step.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 1 |  
+-----+
```

TEXT: THIS REPORT IDENTIFIES SEVERAL POSSIBLE ERROR  
CONDITIONS WHICH STEM FROM PROBLEMS IN THE VCC  
COLLECTION PROCESS --- OR --- DFSMS FLAG  
INCONSISTENCIES AT THE VOLUME OR THE DATA SET  
LEVEL.

FOR MESSAGE GROUP VCA00602W, EITHER:

- (1) SMS MANAGED VOLUME CONTAINS DATA SETS WHICH ARE NOT SMS-MANAGED (MISSING SMS CELL)  
OR
- (2) FORMAT-1 DSCB FIELDS RESERVED FOR USE BY DFSMS HAVE BEEN USED BY NON-IBM CODE RESULTING IN A FALSE INDICATION OF AN SMS-MANAGED DATA SET.

ACTION: OBSERVATION IS WRITTEN WITH SMS DATA ELEMENTS MISSING

FOR MESSAGE GROUP VCA00607W:

- (1) VCC DATA COLLECTION FOR THE SPECIFIED VOLUME DID NOT COMPLETE PROPERLY. THE CAUSE IS SHOWN IN THE VCCSTATS REPORT.

ACTION: OBSERVATION IS DISCARDED FROM THE VCADAA FILE (SINCE VCAVOA OBSERVATION IS MISSING).

TYPE: Information

REASON: This report title is issued for the reasons described in the message title.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 2 |  
+-----+
```

TEXT: >>>> VOLUME=%VOLSER - SYSID=%SYSID %WRKENDTS -  
ERRORS MATCHING VVDS (DFSMS CELL) DATA TO VTOC

TYPE: Warning

REASON: Errors were encountered during the matching of SMS information found in the VVDS to the SMS information found in the FORMAT1 DSCB (VTOC information). A message VCA00603W or VCA00604W will follow.

ACTION: Refer to VCA00603W or VCA00604.

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 3 |  
+-----+
```

TEXT: %DAADSTYP %DSNAME DFSMS INFORMATION CELL MISSING

TYPE: Warning

REASON: The DFSMS information cell in the VVDS is missing. This could be because the data set being flagged as a SMS managed data set in the VTOC is not really a SMS managed data set, but the FORMAT-1 DSCB fields reserved for use by DFSMS have been used by non-IBM code, resulting in a false indication of an SMS-managed data set. This message may be issued if the catalog portion of the scan is turned off by specifying VSAMSUBT=0.

ACTION: The DFSMS fields will be missing in the VCA files.

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 4 |  
+-----+
```

TEXT: %DAADSTYP %DSNAME VTOC REC. NOT FLAGGED  
AS A MANAGED DATA SET

TYPE: Warning

REASON: SMS information was found in the VVDS for a data set. However, there is no SMS information for this data set in the VTOC.

ACTION: Check to see if DAASMSID is turned off in VCAGENIN.

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 5 |  
+-----+
```

TEXT: %DAADSTYP %DSNAME VOL. PROCESSING FAILED  
- THIS RECORD IGNORED

TYPE: Warning

REASON: This message is issued when the volume record is not found. This can mean that some kind of failure occurred while the Space Collector was trying to read the volume.

ACTION: Check the output of VCCSTATS message output for the Space Collector scan job.

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 6 |  
+-----+
```

TEXT: %DSNAME MANAGED DATA SET IN VVDS BUT  
NOT PRESENT IN VTOC

TYPE: Warning

REASON: This message is issued when the volume record is present and the VVDS record is present but the VTOC record for this data set is not present. This could happen if the volume was being read by VCC and it encounters a problem later on, so the volume is only partially processed resulting in some data sets being missed from the processing.

ACTION: Check the VCCSTATS output in the Space Collector scan job to determine the problem.

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 7 |  
+-----+
```

TEXT: >>>> VOLUME=%VOLSER - SYSID=%SYSID %WRKENDTS -  
VOL. NOT SUCCESSFULLY PROCESSED

TYPE: Warning

REASON: This message is issued when errors are detected in matching the volume records with the catalog and VTOC records.

ACTION: See accompanying messages for action.

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 8 |  
+-----+
```

TEXT: >>>> %COUNT1 DATA SETS IGNORED FROM %COUNT2  
VOLUMES.

TYPE: Warning

REASON: This message is issued after the exception report  
and gives the total data sets whose SMS  
information was ignored due to the various causes  
described in the messages COUNT1 = count of data  
sets and COUNT2 = count of volumes.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 6 0 9 |  
+-----+
```

TEXT: >>>> %NNNNNN DATA SETS DELETED FROM VCADAA VIA  
USRSDAA EXIT.

TYPE: Information

REASON: This message is issued if data sets were deleted  
by means of the USRSDAA exit. This is a user-  
defined exit.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 0 6 2 0 |  
+-----+
```

TEXT: DFHSM-OWNED (ML1) DATA SETS FROM SYSID %SYSID  
ENDTS %WRKENDTS)  
-- HIGH-ORDER DSNAME QUALIFIER DETECTED IS:  
%DSNODE1

TYPE: Information

REASON: When building the VCADAA file, VCA performs

pattern recognition on the DSNAME in search of data set name characteristics that indicate a DFHSM-owned data set migrated to a ML1 volume or a backup version staged temporarily on a ML1 volume.

By doing so, there is a small risk that a user may have inadvertently named a data set so that it is mistaken for a ML-1 resident data set owned by DFHSM. For example, if you allocate:

```
DSN=userid.HMIG.TEST.PLAN.DATA.FINAL
```

This would be mistaken for a ML1 data set because the 2nd qualifier is HMIG and the 3rd qualifier starts with the letter T, and it is 6 nodes in total.

**ACTION:** When VCA finds a data set meeting the criteria of a ML1 data set, it issues this message showing the first-level (high-order) qualifier it found for the data set. This value should correspond to the MIGRATEPREFIX or the BACKUPPREFIX parameters of the SETSYS command as established by the DFHSM systems administrator. It is usually HSM or DFHSM but can vary from company to company. The important point is that it is a constant for any given DFHSM implementation.

Therefore, you should have only one of these messages. If you have more than one, it is likely that a user data set has a pattern that is close to that of DFHSM and these data sets should be renamed.

**REFERENCES:** DFHSM Version 2.5 System Programmer's Command Reference SH35-0084

DFHSM Version 2.5 System Programmer's Guide  
SH35-0085

```
+-----+
| V C A 0 1 3 0 0 |
+-----+
```

**TEXT:** THE FOLLOWING FILE SYSTEMS AND THEIR DIRECTORY ENTRIES WERE REJECTED BECAUSE OF INCOMPLETE SCAN.

**TYPE:** Information

REASON: This report will be printed as part of the MICSL0G if the Space Collector fails to completely process a file system and all its directory entries. A subtype 3 record is written after successfully collecting a file system and all its directory entries. If either the file system record (subtype 8) or its corresponding subtype 3 record is missing, a problem in the scan is indicated and the file system is rejected. The file system name is displayed in the report. This is an informational report and shows that the normal error detection and restart process is working correctly.

ACTION: The USS File Systems Collector output should be examined to determine the problem.

REFERENCES: None

```
+-----+  
| V C A 0 1 3 0 1 |  
+-----+
```

TEXT: %DSNAME

TYPE: Information

REASON: This message is issued for every file system that is not successfully processed by the USS File Systems Collector. Information collected by the file system or any of its directory entries will not be written to the HFSDIR file.

ACTION: The USS File Systems Collector's output should be examined to determine the problem.

REFERENCES: None

```
+-----+  
| V C A 0 1 3 0 2 |  
+-----+
```

TEXT: -----  
TOTAL FILE SYSTEMS REJECTED=%MTYP7DEL',  
-----

TYPE: Information

REASON: This message precedes messages VCA1300 and VCA1301 mentioned above. It shows a total count for the file systems that were not scanned successfully.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 1 3 0 3 |  
+-----+
```

TEXT: -----  
TOTAL DIRECTORY ENTRIES REJECTED=%MTYP8DEL',  
-----

TYPE: Information

REASON: This message follows message VCA1304 mentioned below. It shows a total count for the file systems directory entries that were not scanned successfully.

ACTION: None

REFERENCES: None

```
+-----+  
| V C A 0 1 3 0 4 |  
+-----+
```

TEXT: SYSID=%SYSID,RSRCEID=%RSRCEID,HFSRUN=%HFSRUN,VOLSER=%VOLSER,TASKID=%TASKID

TYPE: Information

REASON: This message is issued for every file system's directory entry that has not been successfully processed by the USS File Systems Collector. The file system or any of its directory entries information collected will not be written to the HFSDIR file.

ACTION: None

REFERENCES: None

```
+-----+
```

| V C A 0 1 3 1 8 |

+-----+

TEXT: >>>> HFS DATA IS DETECTED BUT NOT PROCESSED  
BECAUSE NOHFS OPTION IS SET IN VCAGENIN.  
>>>> -----  
>>>> HFS RECORDS READ FROM INPUT DATA = %HFSIN

TYPE: Information

REASON: This message is issued if HFS data is read during  
DAY090 processing, but the HFS option in VCAGENIN  
is set as NOHFS. This option controls whether the  
USS Directory Entry (HFSDIR) file will be built or  
ignored. The code to process HFS data is bypassed.

ACTION: Since HFS data is collected by executing the USS  
File Systems Collector, which is independent of  
the CA MICS Space Collector (VCC), you should  
determine whether the HFS data should be  
processed.

REFERENCES: None

+-----+

| V C A 0 1 3 1 9 |

+-----+

TEXT: >>>> %RECXDIR DATA SETS DELETED FROM HFSDIR VIA  
USRSDIR EXIT.

TYPE: Information

REASON: This message is issued if data sets were deleted  
by means of the USRSDIR exit. This is a user-  
defined exit.

ACTION: None

REFERENCES: None

## A.2 HSM Component Messages

This appendix lists all messages generated by the HSM component of the CA MICS Space Analyzer. Some messages are generated during the processing of the control statements, while others are caused by various conditions in the data found during the processing. The messages are listed in ascending numerical sequence and include the full text of the message, the type, the reason for the message, appropriate user action, and applicable references to documentation.

The following type codes are used to categorize the messages:

Information	designates a note that documents a CA MICS option or potentially important feature in the data.
Warning	designates a condition in either the data or the control statements that does not affect the CA MICS Space Analyzer's operation, but that may lead to unexpected results.
Error	designates that a problem has been encountered with a control statement that will prevent a successful run of the product. Execution is stopped after all control statements are processed.

The message text often includes references to information that is contained in a control statement or values of permanent or temporary variables created during CA MICS processing. In the description below, the message text contains a word beginning with a percent sign (%), which indicates that a value will be substituted into the text at execution time. For example, message HSM00016 reads as follows:

```
TEXT: >>>>> SYSID NOT 1 TO 4 CHARACTERS IN LENGTH.
      SYSID=%TOKEN2.
```

A possible version of the printed message in the MICSLOG might be:

```
TEXT: >>>>> SYSID NOT 1 TO 4 CHARACTERS IN LENGTH.
      SYSID=SYS008.
```

```
+-----+
| H S M 0 0 0 1 |
+-----+
```

TEXT: HSM PARAMETER GENERATION STARTED.

TYPE: Information

REASON: This message indicates the start of the HSMPGEN process.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 0 0 2 |  
+-----+
```

TEXT: HSM PARAMETER GENERATION COMPLETED SUCCESSFULLY.

TYPE: Information

REASON: This message indicates the completion of the HSMPGEN process.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 0 1 0 |  
+-----+
```

TEXT: ---- HSMOPS MEMBER:

TYPE: Information

REASON: This message precedes the printing of the contents of the HSMOPS member.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 0 1 1 |  
+-----+
```

TEXT: %LINE

TYPE: Information

REASON: This message lists the contents of the HSMOPS member.

ACTION: None

REFERENCES: None

```
+-----+
| H S M 0 0 0 1 5 |
+-----+
```

TEXT: >>>> KEYWORD NOT RECOGNIZED KEYWORD=%TOKEN1.

TYPE: Error

REASON: An invalid keyword has been specified in HSMOPS.

ACTION: Specify a valid keyword and resubmit HSMPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| H S M 0 0 0 1 6 |
+-----+
```

TEXT: >>>> SYSID NOT 1 TO 4 CHARACTERS IN LENGTH.  
SYSID=%TOKEN2.

TYPE: Error

REASON: The SYSID value specified on the OPTIONS statement is greater than 4 characters in length. SYSID must be between 1 and 4 characters in length.

ACTION: Specify a valid value for SYSID and resubmit HSMPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| H S M 0 0 0 1 7 |
+-----+
```

TEXT: >>>> INTERVAL TIME VALUE MUST BE 1-999 HOURS.  
VALUE=%TOKEN3.

TYPE: Error

REASON: The interval time specified on the OPTIONS statement is not between 1 and 999.

ACTION: Specify a value between 1 and 999 and resubmit HSMPGEN.

REFERENCES: Section 7.3.2

```
+-----+  
| H S M 0 0 0 2 0 |  
+-----+
```

TEXT: ---- END OF HSMOPS MEMBER.

TYPE: Information

REASON: This message is issued after the contents of HSMOPS has been printed.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 0 2 4 |  
+-----+
```

TEXT: OPTIONS STATEMENT MISSING. ASSUMING DURATION OF 24 HOURS AND ACCEPTANCE OF INPUT FROM ANY SYSID.

TYPE: Information

REASON: The options statement was not found in HSMOPS. A default interval of 24 hours is used and the HSM component will process data from any SYSID.

ACTION: None

REFERENCES: Section 7.3.2

```
+-----+  
| H S M 0 0 0 2 7 |  
+-----+
```

TEXT: ACCOUNTING STATEMENT MISSING. ASSUMING COMPUTATION OF STORAGE COST AT -DAYS- TIME SPAN.

TYPE: Information

REASON: The ACCOUNTING statement has not been specified in HSMOPS. If you use the CA MICS Accounting and Chargeback Option, the storage cost computation for the HSM component will be performed at the DAYS timespan.

ACTION: None

REFERENCES: Section 7.3.2

```
+-----+
| H S M 0 0 0 2 8 |
+-----+
```

TEXT: INVALID PARAMETER FOR %TOKEN1 KEYWORD. SPECIFY %CHOICE1 OR %CHOICE2.

TYPE: Error

REASON: For the ACCOUNTING statement, the valid keywords are DETAIL and DAYS.

ACTION: Specify either DETAIL or DAYS on the ACCOUNTING statement and resubmit HSMPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| H S M 0 0 0 3 0 |
+-----+
```

TEXT: >>>> ERROR(S) ENCOUNTERED IN HSMOPS PARM DEFINITIONS.  
>>>> \$HSMSTR GENERATION ABORTED.  
>>>> CORRECT ERROR(S) AND EXECUTE AGAIN.

TYPE: Error

REASON: One or more errors were encountered during the HSMPGEN process.

ACTION: Correct the errors and resubmit HSMPGEN.

REFERENCES: None

```
+-----+
| H S M 0 0 1 0 0 |
+-----+
```

TEXT: ----- BEGIN PHASE 1 -----

TYPE: Information

REASON: This message is issued during Phase 1 of the DAY095 step of the DAILY job.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 1 0 1 |  
+-----+
```

TEXT: AT LEAST ONE RECORD WITH SYSID NOT DEFINED IN HSMOPS (%ORGSYSID).  
-- THE DEFAULT DURATION FROM PARMS(HSMOPS) WILL BE USED  
-- WHEN AN HSM INVENTORY IS COLLECTED FOR THE FIRST TIME.

TYPE: Information

REASON: An undefined SYSID has been encountered during the processing of the DAY095 step. A default duration value from HSMOPS will be assigned.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 1 0 2 |  
+-----+
```

TEXT: AT LEAST ONE RECORD READ WITH BAD VCH RECORD ID. (%VCHID). SUCH RECORDS ARE DELETED.

TYPE: Warning

REASON: Records were found with a record ID that is invalid. The record ID must be VCH. The records are deleted, and the option-deleted counter is incremented.

ACTION: Check to see if invalid data is being input to the DAY095 step.

REFERENCES: None

```
+-----+  
| H S M 0 0 1 0 3 |  
+-----+
```

TEXT: VCH VERSIONS PRIOR TO %VER ARE NO LONGER SUPPORTED

TYPE: Error

REASON: The input data being processed is from a version of the HSM data collector that is not supported.

ACTION: Ensure that the version of the HSM data collector being used is supported.

REFERENCES: None

```
+-----+  
| H S M 0 0 1 0 4 |  
+-----+
```

TEXT: FIRST INPUT TIMESTAMP IS FROM SYSID (%ORGSYSID)  
-- VERSION (%VER) DATA CREATED ON %ENDTS.  
-- COMPUTED DURATION = %TIME  
-- BASED ON A PREVIOUS COLLECTION ON %PREVTS

TYPE: Information

REASON: The message identifies the first input timestamp, SYSID, version of the data, computed duration, and the last time the collection was done.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 1 1 0 |  
+-----+
```

TEXT: HSM INVENTORY TRAILER RECORD <<%COND>> FOR  
SYSID (%ORGSYSID)  
-- VERSION (%VER) DATA CREATED ON %ENDTS.

TYPE: Information

REASON: This message flags the presence or absence of the HSM inventory trailer record. The value of the variable COND can take on the values of FOUND or MISSING. If the trailer record is missing, then HSM00116W is issued. This may cause input data to be deleted.

ACTION: If the HSM inventory trailer record is missing, then refer to HSM00205W.

REFERENCES: None

```
+-----+  
| H S M 0 0 1 1 1 |  
+-----+
```

TEXT: RECORD-TYPE CODE COUNT  
INCOMING

TYPE: Information

REASON: This message is issued when the HSM inventory trailer record has been found. This message is followed by HSM00112I.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 1 1 2 |  
+-----+
```

TEXT: -- %RECTYPE %RECCODE %RECCNT %ENDTS  
%COND

TYPE: Information

REASON: This message is issued when the HSM inventory trailer record has been found.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 1 1 6 |  
+-----+
```

TEXT: DATA WITH THIS ENDTS WILL BE PURGED TO AVOID  
DUPLICATION  
-- BY A LATER RERUN OF A COMPLETED INVENTORY.

TYPE: Warning

REASON: A HSM inventory trailer record has not been found  
for the data being processed. This usually means  
that a partial HSM inventory took place. These  
records are purged as they are from a partial  
inventory to avoid duplicate data situations.

ACTION: See HSM00205I for action that may be needed.

REFERENCES: None

```
+-----+  
| H S M 0 0 2 0 0 |  
+-----+
```

TEXT: ----- BEGIN PHASE 2 -----

TYPE : Information

REASON: This message is issued during Phase 1 of the  
DAY095 step of the DAILY job.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 2 0 5 |  
+-----+
```

TEXT: FOLLOWING RECORDS PURGED FROM (%FILE) FILE -  
SYSID (%SYSID)  
-- INVENTORY FROM %ENDTS IS BEING DISCARDED.

TYPE: Warning

REASON: Records have been purged from the indicated file  
because the complete inventory has not been input  
to the DAY095 step. This situation happens for  
two main reasons:

- o If the CA MICS HSM component records are being  
written to SMF and an unexpected failure caused  
the Space Collector scan job to abend, a partial

output is written to SMF. When the job is restarted and completes successfully, a complete output is produced. If both the partial output and the complete output are input to the DAY095 step, this message will be issued.

- o If the Space Collector job is running and SMF dumping takes place, only a partial output is written to tape. The rest of the output will be written the next time SMF dumping is done. This will cause the DAY095 to reject these records.

**ACTION:** If the second scenario is the reason for this message, then schedule the Space Collector job so that the HSM inventory completes before SMF dumping is done at your site.

**REFERENCES:**

```
+-----+
| H S M 0 0 2 0 6 |
+-----+
```

**TEXT:** D A T A S E T N A M E

**TYPE:** Information

**REASON:** Appears before HSM00207I and is preceded by HSM00205I

**ACTION:** None

**REFERENCES:** None

```
+-----+
| H S M 0 0 2 0 7 |
+-----+
```

**TEXT:** %DSNAME

**TYPE:** Information

**REASON:** The data set record is being purged from the HSM file.

**ACTION:** None

**REFERENCES:** None

```
+-----+  
| H S M 0 0 2 0 8 |  
+-----+
```

TEXT: ...SUPPRESSING FURTHER DISPLAY OF REJECTED  
RECORDS.....

TYPE: Information

REASON: This message is issued after the limit of  
HSM00207I is reached. The current limit is set to  
40.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 2 0 9 |  
+-----+
```

TEXT: >>>> %RECNUM RECORDS DELETED FROM %FILE.

TYPE: Information

REASON: The total number of records deleted from the file  
indicated is given.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 2 1 6 |  
+-----+
```

TEXT: VOLSER COLLECTION-DATE

TYPE: Information

REASON: Appears before HSM00207I and is preceded by  
HSM00205I.

ACTION: None

REFERENCES: None

```
+-----+  
| H S M 0 0 2 1 7 |  
+-----+
```

+-----+

TEXT: %VOLSER %ENDTS

TYPE: Information

REASON: This message indicates the Volume record that is being purged from the file indicated in the HSM00209I message.

ACTION: See HSM00205I for action to take.

REFERENCES: None

+-----+

| H S M 0 0 3 0 0 |

+-----+

TEXT: SORTING DETAIL DATA INTO DAYS TIME-SPAN SEQUENCE  
FOR FILE:

-- %WRKNAME

-- (EITHER) DAYS SEQUENCE IS NOT A SUBSET OF  
DETAIL.

-- (OR) MULTIPLE ENDTS VALUES ARE PRESENT IN  
THE INPUT DATA.

TYPE: Information

REASON: The sort sequence for the DAYS file is not a subset of the DETAIL file, or there are multiple ENDTS values in the data. This means a sort has to be performed. If the sort sequence for the DAYS file is a subset of the DETAIL file, then this sort can be avoided, thus speeding up DAY095 processing.

ACTION: None

REFERENCES: None

+-----+

| H S M 0 0 9 9 9 |

+-----+

TEXT: ----- BEGIN SUMMARIZATION AND AGING -----

TYPE: Information

REASON: This message indicates that the summarization

process that summarizes the data into higher  
timespans is starting.

ACTION: None

REFERENCES: None



# Appendix B: DATA DICTIONARY

---

The Data Dictionary is only available at your site, where it has been customized to your configuration and your product change level.

To see the Data Dictionary at your site, follow the instructions under Document Browse in the Document Access Guide.